2019

High-Dimensional Design Evaluations For Self-Aligning Geometries

Nicholas Edward Eckenstein
University of Pennsylvania

High-Dimensional Design Evaluations For Self-Aligning Geometries

Abstract

Physical connectors with self-aligning geometry aid in the docking process for many robotic and automatic control systems such as robotic self-reconfiguration and air-to-air refueling. This self-aligning geometry provides a wider range of acceptable error tolerance in relative pose between the two rigid objects, increasing successful docking chances. In a broader context, mechanical alignment properties are also useful for other cases such as foot placement and stability, grasping or manipulation. Previously, computational limitations and costly algorithms prevented high-dimensional analysis. The algorithms presented in this dissertation will show a reduced computational time and improved resolution for this kind of problem.

This dissertation reviews multiple methods for evaluating modular robot connector geometries as a case study in determining alignment properties. Several metrics are introduced in terms of the robustness of the alignment to errors across the full dimensional range of possible offsets. Algorithms for quantifying error robustness will be introduced and compared in terms of accuracy, reliability, and computational cost. Connector robustness is then compared across multiple design parameters to find trends in alignment behavior. Methods developed and compared include direct simulation and contact space analysis algorithms (geometric by a 'pre-partitioning' method, and discrete by flooding). Experimental verification for certain subsets is also performed to confirm the results. By evaluating connectors using these algorithms we obtain concrete metric values. We then quantitatively compare their alignment capabilities in either SE(2) or SE(3) under a pseudo-static assumption.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Mechanical Engineering & Applied Mechanics

First Advisor

Mark H. Yim

Subject Categories

Applied Mathematics | Mechanical Engineering | Robotics

HIGH-DIMENSIONAL DESIGN EVALUATIONS FOR SELF-ALIGNING CONNECTOR GEOMETRIES

Nick Eckenstein

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2019

Supervisor of Dissertation

Mark Yim
Professor of Mechanical Engineering and Applied Mechanics

Graduate Group Chairperson

Jennifer Lukes
Professor of Mechanical Engineering and Applied Mechanics

Dissertation Committee

Vijay Kumar, Professor of Mechanical Engineering and Applied Mechanics

Robert Ghrist, Professor of Electrical and Systems Engineering and Mathematics

Cynthia Sung, Assistant Professor of Mechanical Engineering and Applied Mechanics

Mark Yim, Professor of Mechanical Engineering and Applied Mechanics

HIGH-DIMENSIONAL DESIGN EVALUATIONS FOR SELF-ALIGNING CONNECTOR GEOMETRIES

Acknowledgements

Thanks to my dissertation advisor Prof. Mark Yim for his insightful guidance and steady support. Thanks to the many members of ModLab at Penn over multiple years: Jimmy Sastra, Paul White, Chris Thorne, Matt Piccoli, Kevin Galloway Jay Davey, Jimmy Paulos, and Tarik Tosun. Special thanks are needed for Kevin Galloway, without whose support I would likely have never reached this point. Lastly, thanks to my family for their support throughout my time at Penn. Special thanks to Sonia Roberts for her support as well.

ABSTRACT

HIGH-DIMENSIONAL DESIGN EVALUATIONS FOR SELF-ALIGNING CONNECTOR GEOMETRIES

Nick Eckenstein

Mark Yim

Physical connectors with self-aligning geometry aid in the docking process for many robotic and automatic control systems such as robotic self-reconfiguration and air-to-air refueling. This self-aligning geometry provides a wider range of acceptable error tolerance in relative pose between the two rigid objects, increasing successful docking chances. In a broader context, mechanical alignment properties are also useful for other cases such as foot placement and stability, grasping or manipulation. Previously, computational limitations and costly algorithms prevented high-dimensional analysis. The algorithms presented in this dissertation will show a reduced computational time and improved resolution for this kind of problem.

This dissertation reviews multiple methods for evaluating modular robot connector geometries as a case study in determining alignment properties. Several metrics are introduced in terms of the robustness of the alignment to errors across the full dimensional range of possible offsets. Algorithms for quantifying error robustness will be introduced and compared in terms of accuracy, reliability, and computational cost. Connector robustness is then compared across multiple design parameters to find trends in alignment behavior. Methods developed and compared include direct simulation and contact space analysis algorithms (geometric by a 'pre-partitioning' method, and discrete by flooding). Experimental verification for certain subsets is also performed to confirm the results. By evaluating connectors using these algorithms we obtain concrete metric values. We then quantitatively compare their alignment capabilities in either SE(2) or SE(3) under a pseudo-static assumption.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modular self-reconfigurable robotic systems hold the promise of low cost, versatility, and robustness. The aim of research on these types of systems is to find a sort of 'universal robot' which can perform any task simply by reconfiguring itself. Modular self-reconfigurable robots, however, present a wide array of challenges in terms of design and implementation.

A **modular** robot is defined as one that is "built from several physically independent units that encapsulate some of the complexity of their functionality" [63]. A reconfigurable modular robot is defined as one where the module's connectivity can be changed. These systems are classified according to three criteria: *lattice*-style denoting systems which nominally sit on a lattice and reconfigure by changing connections with neighbors, *chain*-style denoting systems which form and reform into chains, and *mobile* denoting systems which are capable of movement independent of other modules [64]. Reconfiguration for each of these systems is accomplished by repetition of a process known as docking, the physical connection of two modules. One of the key challenges in modular self-reconfigurable robotics is ensuring that this docking process is stable and reliable, without requiring too much design overhead.

In *chain*-style modular robots in particular, positioning errors accumulate exponentially for the number of modules in the chain. These errors must be corrected by the system if it is to dock and reconfigure successfully.

The docking process is divided into two parts: *alignment*, in which the modules obtain a sufficiently accurate relative position, and *latching* where the physical connection between the modules is made or forces put in place which maintain the alignment.

In general, docking between two modules occurs at special docking faces on a module also called a connector. Faces can be gendered or ungendered. Gendered faces have male and female features which must be matched, and thus must be carefully controlled to ensure connections are formed from one of each face. Ungendered faces do not have this restriction. Hermaphroditic faces have both male and female features on each face. Whereas single gendered faces can only mate with their opposite gender, hermaphroditic and/or ungendered faces can mate with any other face.

Another requirement for reconfigurable connectors is the ability to undock successfully as well as dock. If mechanical connection is being used, either the docking or the undocking requires some actuation capabilities. One or the other of these processes can proceed passively but in order for the connection to be stable under all possible force/torque conditions, there needs to be an active state change. Historically this has been accomplished by Shape Memory Alloy (SMA) or small motors [34] [75].

Beyond modular reconfigurable robots, the docking problem has applications to many different problems in controls and robotics. Quick-change end-effectors and pick-and-place operations in automated assembly lines often have to solve the related 'peg-in-hole' problem [14]. Mid-air plane refueling makes use of a conical 'probe-and-drogue' shape to ensure that slight misalignment between aerial vehicles does not result in a failed refuling operation [71].

Robot assembly has long made use of 'remote center of compliance' (RCC) devices at the end effector to ensure good alignment between parts during insertion tasks. These devices simulate rotational and translational springs at the point of insertion of parts, by use of multiple flexible components in the end effector [13]. This allows for more beneficially flexible motion of the part in response to the surface it contacts, resulting in greater alignment abilities. Design and analysis of these type of systems are well understood [15], and reveal that the location of the center of compliance is an important factor to consider in alignment problems. Variable RCC systems have been developed using elastomer shear pads and insertion of a stiffness adjusting rod to control the stiffness level [38]. Compliance at the end effector can also be added by using soft servos, though the evidence suggests it may be inferior to force control methods for insertion tasks [12].

In order to build the most capable robotic systems possible, we seek as our main goal to maximize the allowable error in position and orientation that two connecting members can have yet still dock correctly. Satisfactory multi-dimensional metrics for this error do not yet exist, nor do practical methods by which to determine the ranges of these errors. New metrics should fairly balance position errors with orientation errors, as well as attempt to find a good match between a robotic system's likely errors and the errors we can correct. New methods for evaluating the range should take into account not only approach angle between connectors but also their relative compliance and the behavior and possible outcomes that follow during contact. Accomodations and adjustments should be made to allow for fair comparison between connector geometries of different height and width. Once practical metrics and methods exist, we would look to identify trends in connector design that will help us maximize the allowable error. Finally, a practical connector should be constructed and tested to prove the viability of the method.

In this dissertation, we present metrics, assumptions, and techniques to capture the capabilities of these geometries and answer the above questions. We then use our

metrics to compare different types of self-aligning geometries, and describe patterns in design parameters which can lead to improved self-aligning geometries. Finally, we inform the development of a docking mechanism which passively establishes a physical connection without interfering with the self-alignment of the geometry.

1.1 Connection Types

Essential to the act of reconfiguration is the mechanism by which modules are physically mated together. There are many different ways to characterize these connection mechanisms. Table 1.1 shows many of the MRR systems and their connection mechanisms.

Several qualitative terms used here to categorize these connectors are explained below as first introduced in [22].

Self-Aligning Degree represents the degree to which the connector *passively* aligns the two faces, such as magnetic or mechanical forces. A 'High' Rating indicates self-alignment capability in one offset direction approximately greater than 20% of the characteristic size of the module face. 'Low' rated systems have some self-alignment capability but less than 20%. 'None' rated systems have no self-alignment capabilities and must be aligned carefully either by active robotic mechanism or by hand.

Gendering represents whether connectors are interchangeable or must be paired in a particular manner. Gendered connectors have a 'male' and 'female' face - male faces can only pair with female faces, and vice versa. Ungendered connectors do not have this restriction - any face can pair with any other face.

Connection Activity and **Disconnection Activity** indicate whether the act of connection/disconnection requires an action on the part of a module. **Connection Agency** and **Disconnection Agency** indicate which modules are required to be operational/active for the respective action. Double End Agency requires both

faces to cooperate to accomplish the connection/disconnection, Single End Agency requires only one functioning face (either one), and Male/Female requires the indicated (single) face.

Connection Type indicates the mechanism by which connections are accomplished. Most systems use either magnetic mechanisms or mechanical latching with a few systems using electrostatic forces or pressure to maintain the connection.

Connection Maintenance indicates the extent to which power is required to maintain a connection. Generally speaking, it is undesirable to have a system require power simply to maintain its shape. This is especially true in modular systems which often have a limited power budget.

Compliance indicates the flexibility of the connection. 'Rigid' connections have a mechanically rigid connection between module frames. 'Compliant' systems have some flexibility to external forces, either from springs/compliant parts or magnetic compliance.

Approach Angle indicates the direction of approach that the system most regularly encounters. Systems with a direction of approach perpendicular to the face are generally more responsive to self-alignment design features.

Connector	Month/Year	Self-Aligning Degree	Gendering	Connection Activity	Disconnection Activity	Connection Agency	Disconnection Agency	Connection Type	Connection Maintenance	Compliance	Approach Angle
CEBOT Mark II	May/1989	High	Gendered	Active	Active	Male End	Male End	Latching	Zero-Power	Rigid	Perpendicular
Fracta 2D	May 1994	High	Gendered	Active	Passive	Male End	Male End	Magnet	Powered	Compliant	Non-Perpendicular
Metamorphic	August 1996	High	Gendered	Active	Active	Female End	Female End	Latching	Zero-Power	Rigid	Non-Perpendicular
3-D Unit	May 1998	Low	Ungendered	Active	Active	Double End	Double End	Latching	Zero-Power	Rigid	Perpendicular
Molecules	May 1998	Low	Gendered	Active	Passive	Double End	Single End	Magnetic	Powered	Compliant	Perpendicular
Telecubes	May 1998	Low	Ungendered	Passive	Active	Double End	Single End	Magnetic	Zero-Power	Compliant	Perpendicular
I-Cubes	Apr 2000	None	Gendered	Active	Active	Double End	Double End	Latching	Zero-Power	Rigid	Perpendicular
CONRO	Nov 2001	Low	Gendered	Passive	Active	Female End	Female End	Latching	Zero-Power	Rigid	Perpendicular
DRAGON	Dec 2002	High	Ungendered	Passive	Active	Double End	Double End	Latching	Zero-Power	Rigid	Perpendicular
Polybot G3	March 2003	Low	Ungendered	Passive	Active	Double End	Double End	Latching	Zero-Power	Rigid	Perpendicular
Crystalline	Sep 2003	Low	Gendered	Active	Active	Male End	Male End	Latching	Zero-Power	Rigid	Non-Perpendicular
ATRON	Oct 2004	None	Gendered	Active	Active	Male End	Male End	Latching	Zero-Power	Rigid	Non-Perpendicular
amour	Apr 2005	High	Gendered	Active	Active	Female End	Female End	Latching	Zero-Power	Rigid	Perpendicular
Slimebot	Apr 2005	None	Gendered	Passive	Passive	Single End	Single End	Velcro	Zero-Power	Compliant	Perpendicular
GHC (Catom)	May 2005	None	Ungendered	Active	Passive	Double End	Double End	Electrostatic	Low Power	Compliant	Non-Perpendicular
Molecubes	August 2006	Low	Ungendered	Passive	Active	Single End	Single End	Magnetic	Zero-Power	Compliant	Perpendicular
CKBot (SAE)	Oct 2007	Active	Ungendered	Passive	Manual	Double End	Double End	Magnetic	Zero-Power	Compliant	Perpendicular
M-TRAN III	Mar 2008	None	Gendered	Active	Active	Male End	Male End	Latching	Zero-Power	Rigid	Perpendicular
Roombots	May 2008	None	Ungendered	Active	Active	Male End	Male End	Latching	Zero-Power	Rigid	Non-Perpendicular
SINGO	May 2009	High	Ungendered	Active	Active	Single End	Single End	Latching	Zero-Power	Rigid	Perpendicular
RATChET	Sep 2009	Low	Gendered	Active	Passive	Single End	-	Magnetic	Zero-Power	Rigid	Non-Perpendicular
Pebbles	May 2010	None	Ungendered	N/A	Active	N/A	Single End	Magnetic	Zero-Power	Compliant	N/A
Vacuubes	May 2010	High	Ungendered	Active	Passive	Single End	Single End	Air Pressure	Single Pump per Cluster	Rigid	Perpendicular
DFA	Sep 2011	High	Ungendered	Passive	Passive	-	-	Magnetic	Zero-Power	Rigid	Perpendicular
SMORES	Oct 2012	Low	Gendered	Active	Active	Male End	Male End	Mag/Latching	Zero-Power	Rigid	Perpendicular
M-Blocks	Nov 2013	Low	Ungendered	Passive	Passive	-	-	Magnetic	Zero-Power	Compliant	Non-Perpendicular
DARPA TEMP	June 2014	High	Gendered	Active	Active	Double End	Double End	Latching	Zero-Power	Compliant	Perpendicular

Table 1.1: Connector Systems Classification; Various Systems by Year

Chapter 2

Area of Acceptance

When these connectors' relative self-alignment or error correction capabilities are described, they are typically given as one dimension at a time i.e. ($\pm 5mm$, $\pm 8°$). However, it behooves us to look at combined errors in multiple dimensions as is often true in real-world applications.

In order to be more precise about the offset positions which result in successful alignment, we have defined the term **area of acceptance**. Area of acceptance is defined as "the range of possible starting conditions for which mating will be successful" [20]. Practically speaking, what this means is that if the docking procedure is executed given some initial misalignment offset between connectors, the alignment features of the connector will correct the offset. The range over which this occurs is the area of acceptance. Note that while we call this 'area' of acceptance, in higher dimensions it is actually a volume or hypervolume. Area of acceptance can be difficult to determine; for three-dimensional systems it contains two positional offsets and three orientation offsets (we consider all points along the approach direction to be the same, removing one translational DOF). Area of acceptance is always scaled to the characteristic width of the connector D in translational dimensions, and has a range from $\frac{\pi}{2}$ to $\frac{\pi}{2}$ in rotational dimensions. This ensures fair comparison between connector *geometries* - it would not be fair to compare larger connectors of one

geometry to smaller connectors of another.

The concept of area of acceptance is important because alignment and connection systems need to be error-tolerant in order to be successful. Long module chains in particular have a tendency to accumulate errors quickly resulting in failed connections if connectors are not sufficiently corrective. These errors could be in multiple dimensions at once, so it is best to measure the area of acceptance over as many dimensions as is feasible. We seek through multiple methods to determine the conditions under which alignment will be successful, and analyze these conditions to tell us which connectors are most useful under specific or general error conditions.

This chapter is dedicated to explaining what area of acceptance is and how it can be measured. We present several new ways for determining area of acceptance in the following chapters, including novel numerical methods which enable quick analysis of high-dimensional area of acceptance.

2.1 Types of Area of Acceptance

We can classify AA (area of acceptance) domains into different types with respect to the DOFs of one docking element relative to the other. Each DOF is either *constrained, unconstrained*, or the *approach* DOF. In the 2D X-face, for example, the docking elements are constrained to be in a plane, and one face has two translation DOFs and one rotational DOF relative to the other (3 DOF equivalent to $SE(2)$). However, one of those translation DOFs is the mating direction. That DOF can be considered to be constrained while the other two are able to move freely, albeit pseudo-statically (to simplify analysis). In the more general 3D case, one face has six DOFs relative to the other ($SE(3)$) with one DOF as the approach. While this approach DOF is typically a pure translation, it could be any trajectory that leads to lower kinematic pair relationships - including screws. In 3D, subtracting the approach DOF leaves five other DOFs that are unconstrained. For the sake of

clarity, throughout this document we use 'constrained' in the sense that those DOFs will always be locked in the state corresponding to the perfectly aligned state.

In some cases, where the faces have a symmetry (e.g. a round peg in a round hole) one or more DOF may be in a "don't care" state. Whereas the end mating condition in all other cases has the state of all DOFs defined, this one has the other symmetric DOFs make no functional difference and so do not need to match to satisfactorily mate. In these cases we exclude the "don't care" DOF from analysis entirely.

2.1.1 Zero Rotation Area of Acceptance

A simple concept of area of acceptance is called the Zero Rotation Area of Acceptance (ZRAA) which assumes that all rotational degrees of freedom are removed or constrained and the approach direction is perpendicular to the face. Simply put, this means that the orientation is locked for both faces in the correct position, and we are only considering translation errors. For systems in $SE(2)$, this is equivalent to Nilsson's self-alignable offsets [48] - simple analytically defined bounds based on the representation of a 2D connector face as a function. In the 3D/$SE(3)$ case it represents a two-dimensional set of points (over x and y in the plane). ZRAA is much easier to compute than other types of AA, and often is immediately obvious from visual inspection of connector geometries. This gives us a relatively simple, quick picture of the acceptance potential of the given connector. Some diagrams of this ZRAA are shown in Figure 2.1, and were determined analytically based on given dimensions of the connectors in question obtained from the literature.

For purely mechanically actuated self-alignment (e.g. no magnets) we have reviewed many connectors from the literature and characterized a large set of active and passive connectors in Table 2.1 as a sum total of the positions corrected normalized with respect to the connector cross-sectional area. These ZRAAs are determined based on the size and shape of the alignment features (pegs and holes, etc.) given by the authors as well.

From Table 1.1, we can see that the first modular robot system to use a 3D self-aligning docking connector is the cone-shaped dock on CEBOT [26], which was a simple cone-and-funnel alignment system. Other systems have used this connector shape since then, including the AMOUR underwater robot system [68], which called it a probe-and-drogue shape since it in fact had a 'probe' end that would be latched into the receptacle on the adjoining module. This gendered connector geometry has a relatively wide ZRAA (see Table 2.1)given the space it takes up, however the cone feature does not align offsets in the *yaw* direction; that is, around the axis of the face. Polybot [74] used a self-aligning connector to bridge the gap over fine resolutions where sensing failed, typically at the end of a long chain where errors accumulate. As a result, the 4 connecting pin/hole pairs on each face cover a very small area compared to the size of the face (4mm pin diameter, 50mm face width). Due to the pairing, this connector is hermaphroditic (containing both male and female parts), allowing faces to connect to any other face. Nilsson [49] was the first to use the geometry of the connector as a design feature for robotics. The DRAGON connector was designed specifically for high-strength, high-acceptance applications, both of which are important for modular robotic applications as the number of modules in the system grows. Nilsson characterizes the viable offsets which the connector is capable of correcting, both linear and angular (15mm=$\varnothing/5$ linearly and 45 deg angular). The DRAGON connector uses 4 cone-funnel pairs, alternating in a circle, with the latch on an outer ring. The values for these and other connectors in this table were determined by measurements given on the alignment geometries either in the text or in figures. Passive mechanisms are analyzed by the areas of the two aligning connector components, and active mechanisms from correction ranges given in the text.

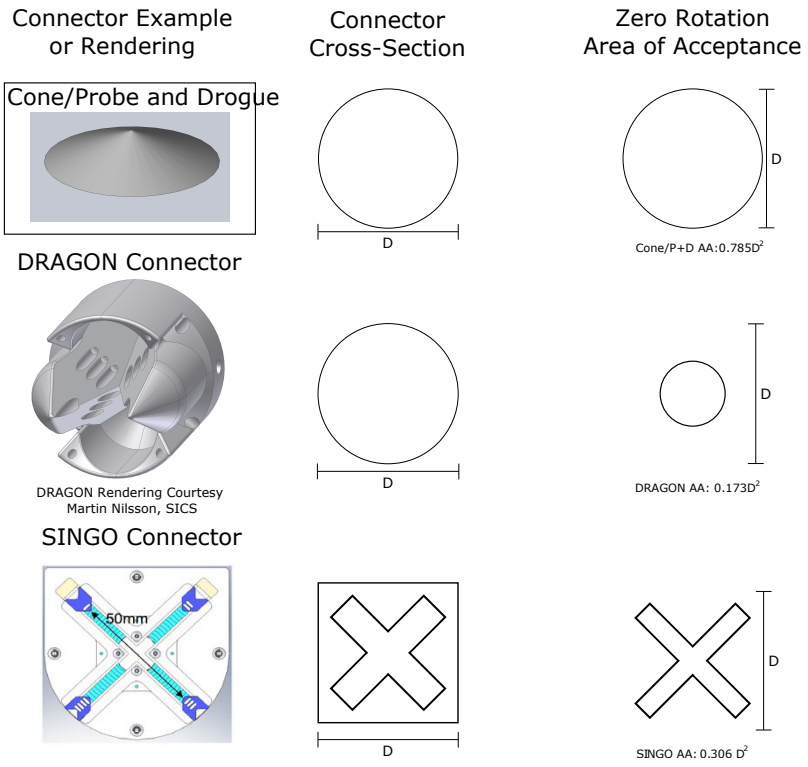
Many modular systems are not included because they are not explicitly comparable; either the connection mechanism is not 3D, or they rely on magnetic forces to align. The systems in the table utilize either active or passive mechanical forces

to align the modules for docking. Connectors utilizing active mechanisms do so as a combination alignment and attachment mechanism. The other named connectors presented here (Cone, Polybot, DRAGON, X-Face) are all passive alignment, allowing us to do remove the design resources required for actuation, power, and control in the connector face.

Also worth noting are the interesting alignment geometries on many of Lipson's robots [27] [76]. These are mechanical alignment faces which augment either a pressure-driven or magnetic alignment force. Here we evaluate only the alignment geometry, not the other forces.

Table 2.1: Table of ZRAA metrics, normalized relative to characteristic length of the face. These are exact where possible from the data available in the literature, otherwise estimated from available data. Entries marked with a * indicate estimated ZRAA rather than exact. Table reproduced from [21]

System	Normalized ZRAA Sum
GENFA Connector	0.00353
Polybot	0.00503
M-TRAN III	0.00592
JHU	0.00592
I-Cubes*	0.0187
CONRO*	0.0425
Vacuubes	0.0555
ACOR(unpaired)	0.0711
SINGO Connector	0.306
DRAGON	0.353
amour	1.57



SINGO Data from [59]

Figure 2.1: Comparison of ZRAAs. The leftmost column shows the connector geometry, the middle column compares connector cross-sections viewed from the approach direction. The rightmost column shows the ZRAA. All values assume the shapes scale to maximum width D to keep comparisons size-invariant

2.1.2 Full Area of Acceptance

In most cases, we want to characterize connectors based on all possible error states. The full AA is the AA given no exclusion of possible starting conditions and no constraints. For 2D connectors the full AA is two-dimensional (x and θ). For 3D connectors it is five-dimensional (x,y,$pitch$,$roll$,yaw). We have not seen any connectors in the literature where the Full AA is calculated. There currently exist no analytic methods to find the Full Area of Acceptance, though we propose several new fully developed numerical methods confirmed by experiment in Chapters 3-5. These methods have some computational limitations, however those limitations vary depending on the desired quality of results.

2.1.3 Other Areas of Acceptance

In order to effectively visualize the Full AA without exhaustively exploring the five dimensional space, we sometimes take two dimensional slices. These slices are obtained by starting with all DOFs aligned, except the two being explored. Those two are sampled at high resolution to obtain an intuitive 2D plot of the AA for those DOFs. Note that the other non-approach DOFs are not constrained in these slices.

These two dimensional subsets we call the X-Y Slice, X-Roll Slice, X-Pitch Slice, and X-Yaw Slice Areas of Acceptance. The X-Roll Area of Acceptance is the subset of the Full AA with initial offsets in the x and *roll* dimensions. The other two areas of acceptance are similarly named. These results still represent a system which is able to move fully in 6DOF, but shows the relationships between two dimensions of offset if the others have none.

In further cases, it is useful to restrict the motion of a system and examine what acceptance is achieved when a 3D system is only capable of motion in a limited set of dimensions eg. (x, y, *roll* (as well as the approach direction z). This kind of area of acceptance is called eg. X-Y-Roll or X-Roll-Pitch, etc. These spaces are very different from the **Slice** AAs, since they exist in $SE(3)$ but are restricted to move

in only the specified dimensions. The **Slice** AAs however, are slices sampled from a space where the connectors are free to move in $SE(3)$.

2.2 Figure of Merit

Now that we have enumerated the possible areas of acceptance in terms of the spaces they occupy, we turn to the methods by which we measure these spaces. Not all measurements are created equal - depending on the relative likelihood of the system errors we might prefer a shape which is smaller but better suits those errors. Aside from the naive measurement (simply summing the points) we present a more refined single-figure quantitative metric appropriate to general robotics platforms. Additionally, we compare these two metrics in this section.

Note that in all cases, the linear dimensions are normalized to the connector size i.e. a width or diameter, to avoid giving an advantage to connectors which are simply larger.

2.2.1 Sum Metric

The **Sum** metric is found by simply adding up the total area that is accepted. This gives a simple figure of merit of all possible configurations without giving any information as to the shape of this area. The Sum Metric with the ZRAA is used in Table 2.1. This metric is reasonably informative for the ZRAA, where concavities are unlikely to occur and shapes are typically not complex.

However, it has been our experience that concavities and irregular shapes do occur, in particular when orientation offsets are introduced. For example, the ZRAA for the SINGO connector, seen in Figure 2.1, takes up a sizeable area, but is not as robust to high error solely in x or solely in y. Additionally, we may find in some instances that while the Sum Metric for a particular connector is reasonably high, it could be highly robust to errors in one dimension, but less robust in another. In

order to better measure the connector acceptance in a way that is more consistent to characteristic robot error, we have developed an alternative figure of merit that reflects error range in each dimension.

2.2.2 Oriented N-Cube Metric

While adding the total area of an AA is useful, it does not always resemble the most practical fit to a practical robotic application. Typically, each dimension has some range over which error ranges can be expected based on the specific robotic platform. One would like to match the positioning error of the robot with the AA. Concavities and narrow areas or holes in an AA can limit this matching - errors are typically convex in nature. If the error space of a specific platform has been characterized to determine the likely error, this could be paired with the AA to determine the likelihood of misalignment. However, it is more likely that this kind of connector-robot matching data is not available since it requires conclusive testing of the error.

Besides the cost of testing the robot error, we seek a generally more accepting geometry for use in multiple applications - the aim is to maximize alignment for all robot platforms by assuming a convex error space with decoupled dimensions. Thus we introduce a new metric that is system-neutral but more representative of capabilities with respect to typical robotic platforms.

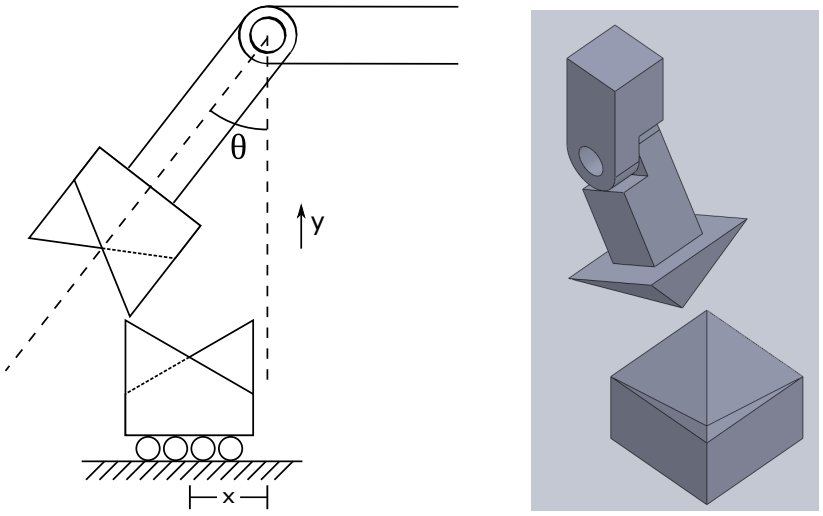
The **Oriented N-Cube Metric**, as a figure of merit is more likely able to match with positioning errors that typically are defined as distances from an ideal position or orientation. We define it to be a measure of the AA found by the edge size (characteristic length) of the largest axis-oriented N-Cube which fits within the normalized AA. An **N-Cube** is a cubic shape existing in N dimensions; e.g. an N-Cube in $\mathbb{R}^2$ is a square, an N-Cube in $\mathbb{R}^3$ is a cube. We only consider N-Cubes which are axis-oriented, meaning that its edges are parallel to the $(x,y,...)$ axes. For the metric we find the largest axis-oriented N-Cube which can fit fully within the

normalized AA; that is, the largest N-Cube which contains only offsets that result in successful alignment. This N-Cube has a diameter (characteristic length) which we then use as our single-number metric. The **normalized AA** is the AA with each dimension divided by its maximum feasible limits. For angular offsets this will typically be $-\pi/2$ to $\pi/2$, and for positional offsets it will be the maximum offset before connectors are no longer touching. So on a connector with maximum positional offset of 5 mm and maximum angular offset of $\pi/2$ radians, the offset (2mm,$\pi/4$ radians) would become (0.4,0.5). Normalizing in this fashion allows us to compare angular and positional offsets and define an N-cube with unit-less measures.

The N-cube metric is also relatively simple to calculate from an AA shape. Implementation details for how to do this depend on the representation, see Section 5.3.1.

As a further example we can see in Fig. 2.1 that the largest oriented N-Cube (or square, in the case of the ZRAA) for the DRAGON or Probe and Drogue would be nearly the same area as the full sum (as the square inscribed in the circular AA), but only a small square can fit in the center of the SINGO ZRAA due to the highly concave 'X' shape. This shows how the shape of the AA can have a large effect on the Oriented N-Cube Metric compared to the Sum Metric.

2.3 Assumptions and Problem Statements



(a) 2D case example, X-Face connector

(b) 3D case example for offsets in (x,y,$pitch$), Pyramid connector

Figure 2.2: Robot and arm connector setups

Alignment in the real world is influenced by many different factors: not only the geometries in question, but also the static and dynamic forces, friction and restitution, and system characteristics. It is important then to ensure that we have defined the exact nature of the problem before starting work. We call the combination of these relevant factors (excepting the geometry, which is the element being compared) the *Alignment Problem Specification (APS)*. As an example, for many 2D connectors' analysis (as in Chapter 3) we look at an idealized scenario. Example Alignment Problem Specifications are presented in Table 2.2 for the two cases where no degrees of freedom are restricted.

If we wish to restrict one or more degrees of freedom as we will later in Section 5, we simply specify the remaining free degrees e.g. Table 2.4.

These restricted spaces are useful if we want to know either how a restricted system would behave, or for some intuitive notion of how a full system might move

Condition	State
Forcing Direction:	$x = x_0$
Frictional:	Zero Friction
Damping:	Critically damped
Restitution:	Zero
Initial Offset Dimensions:	2 linear(x & y) and 1 angular (θ)
Free Dimensions:	x, y, & $theta$

Table 2.2: Simple 2D Alignment Problem Specification

Condition	State
Forcing Direction:	$z = z0$
Frictional:	Zero Friction
Damping:	Overdamped
Restitution:	Zero
Initial Offset Dimensions:	2 linear(x & y) and 3 angular (*roll*, *pitch*, and *yaw*)
Free Dimensions:	2 linear(x & y) and 3 angular (*roll*, *pitch*, and *yaw*)

Table 2.3: Full 3D Alignment Problem Specification used in Gazebo simulations

in the degrees left free, if those are what we care about.

It is important to specify not only the physical assumptions with regards to friction etc. but also the free and fixed dimensions for each of the two parts being aligned. This gives us the clearest possible picture of the problem and avoids confusion that could result from testing connectors under inequivalent conditions.

Problems within this work assume a linear forcing condition perpendicular to the plane of alignment, zero friction and restitution, and critical damping. Diagrams of the physical forcing conditions for a 2D case and a 3D case can be seen in 2.2.

Condition	State
Forcing Direction:	$z = z0$
Frictional:	Zero Friction
Damping:	Overdamped
Restitution:	Zero
Initial Offset Dimensions:	2 linear(x & y) and 1 angular (yaw)
Free Dimensions:	2 linear(x & y) and 1 angular (yaw)

Table 2.4: Alignment Problem Specification for X-Y-Yaw AA, a three-dimensional AA

2.4 Design Parameters Affecting Acceptance

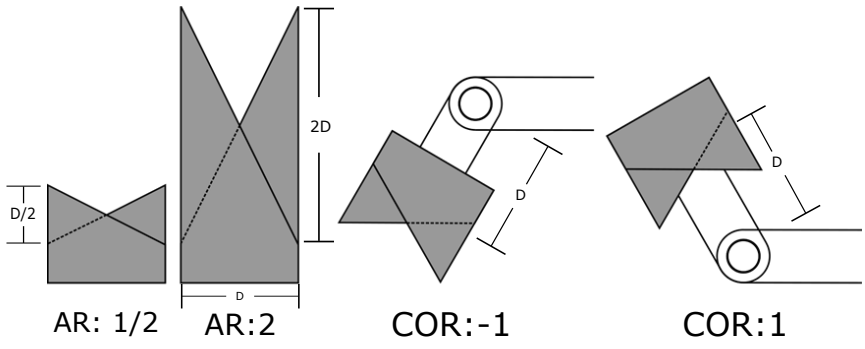


Figure 2.3: Examples of the two design parameters, AR and COR with the 2D X-Face connector [20]. Note that the arm is 'virtual' and would not collide with the other connector.

We have found two relevant design parameters (Fig. 2.3 that significantly affect the area of acceptance, and thus should be taken into consideration when comparing geometries. First the aspect ratio (AR) - measured as the height of the connector geometry H relative to the width of the base D, meaning $AR = 1 = H/D$ is a connector with height equal to its width. For 3D connectors we assume a square base, with width D as one side of the square.

Next is the remoteness of the center of rotation (COR). Center of rotation has a significant effect on the AA. The COR parameter is quantified as the distance from the center of the connector to the rotation point with positive values in 'front' of the connector (more remote) and negative values 'behind' it. These are normalized by the characteristic width of the connector D. For example in Fig. 2.3 the COR value on the 3rd figure is -1, while the COR value on the 4th figure is +1. It is well known from research on the 'peg-in-hole' problem that alignment systems with a remote (positive, as defined above) center of compliance [14] have good passive error correction.

2.5 Combining Areas of Acceptance

In some cases where there are two different connectors with a certain method of connection, we can find the area of acceptance by using convolution to combine two other areas. For example, the DARPA TEMP system [50] has an active gendered connector. The connector is composed of a male hook end and a female loop end. These two connectors each sweep out a separate volume, but only need to cross at one point for connection to be successful. We can then compose the area of acceptance for the two robots by combining the two swept volumes, as a convolution. The convolution of the 2D representation of the two connector areas is shown in Figure 2.5. This area of acceptance was approximately validated by experiment. Some discrepancies exist due to additional dynamic constraints of the robot system and interactions between robots.

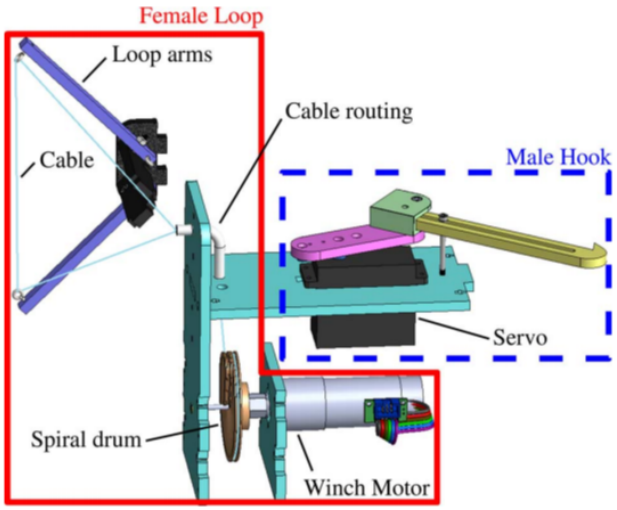


Figure 2.4: Diagram of DARPA TEMP hook and loop docking actuation. Both ends of the loop string are fixed to the one winch, resulting in a single DOF for the female loop connector.

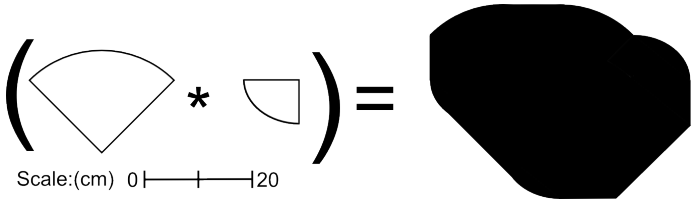


Figure 2.5: The docking connector 2D area of acceptance (right) is obtained by convolution of the shape swept out by the hook (left) and the shape swept out by the loop (center).

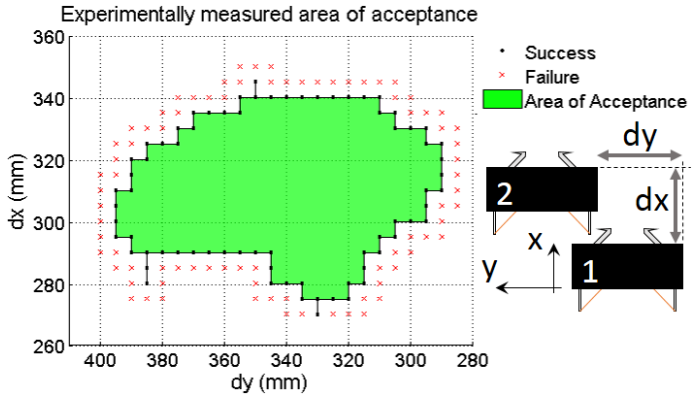


Figure 2.6: Experimental data collected to show the actual 2D area of acceptance (boats not drawn to scale).

Chapter 3

2D Connector Types and Alignment Limits

We begin by examining 2D connectors and positing a new connector geometry, in an effort to find and test a simple method for determining and comparing 2D connectors. Our first method is a simple iterative kinematic analysis intended to step through the alignment process and find the AA for a few 2D connectors. We compare the performance of the new 2D connector to the best-case bounds of other known connectors from the literature to show that the new connector has increased performance over other standard connectors.

Two-dimensional connectors and their limits have been studied briefly in the literature. In [48], Nilsson proves that for single-piece faces (faces which can be written as a function $y = f(x)$ in a plane, where y is the direction of motion to mate two matching faces), a face that gives the largest bi-directional offset correction possible for identical connectors is $D/3$ where $x(y_{max}) = D/3, x(y_{min}) = 2D/3$. He also proves that for gendered faces this relationship is $D/2$ where $x(y_{max})) = D/2$ for the male and correspondingly $x(y_{min}) = D/2$ for the female. We call these the S-Face and the V-Face respectively. The Alignment Problem Specification for Nilsson's analysis is shown in Table 3.1.

Condition	State
Forcing:	$x = x_0$
Frictional:	Zero Friction
Damping:	Critically damped
Restitution:	Zero
Initial Offset Dimensions:	2 linear(x & y)
Free Dimensions:	1 linear(x)
Base Face	Free in x and y

Table 3.1: Nilsson Alignment Problem Specification

Figures 3.1 and 3.2 show the simplest example of the identical faces (left), gendered faces (center) and a newly proposed geometry (right).

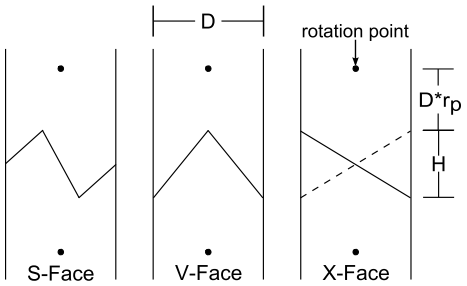


Figure 3.1: A graphical representation of the face geometries. All three faces shown here have the same 'profile area', represented by H and D, the height and width.

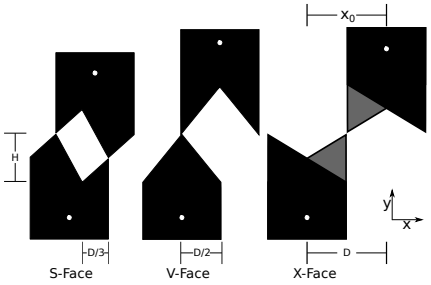


Figure 3.2: A representation of the maximum lateral offset for which self-alignment can occur, for each of the S, V, and X-Faces.

The S-Face is the name we give to the identical face geometry analyzed by Nilsson. In this example we will use the simplest profile of the face (four vertices connected by straight line segments). Other geometries that have the same max and min conditions exist and similar analysis will apply without loss of generality. If the profile is viewed horizontally (as in Fig. 3.2) with horizontal width D, one vertex is at one-third of the way across the face, and the other is at two-thirds. The two middle vertices are symmetrically offset from the vertical, one up and one down.

This shape gives the maximum lateral offset that still yields self-alignment for a 2D ungendered profile [48], an offset of $D/3$ in either direction.

The V-face is the 2D equivalent of the cone and hole shape or probe and drogue as seen in Fig. 2.1, a shape representing the maximal lateral offset for 2D profiles if we allow for the connectors to be gendered. The male profile is essentially an isosceles triangle. The maximum lateral offset is $D/2$.

The X-Face [20] is a new face geometry designed to maximize acceptance range for real-world robots aligning in a SE(2) plane. It is composed of two layers, with the top a mirror image of the bottom, so it is no longer purely 2D, though the motions are still constrained to be planar. These faces will self-align in the lateral case as Nilsson examines it up to the full width D of the connector away, an improvement threefold over the S-face. Additionally, the X-Face does not suffer from the drawback of being

gendered, meaning we do not have to check genders of faces before mating them. Since the self-alignment process is defined by contact of the two pieces, this is the theoretical maximum lateral offset for planar constrained motions of self-alignment.

In the previous case we have assumed: planar constrained motion, the same size (D) for both connectors, no rotations when docking and the vertical axis as the docking direction.

3.1 Simplified Kinematic Analysis

For the 2D faces above, the mating problem can be divided into three phases: the approach phase, the alignment phase, and the slide phase.

During the approach phase, the parts approach along the direction of forcing until the first contact point is made. In the alignment phase, the parts rotate and slide relative to each other, maintaining contact until either the parts are either aligned with the same angle, or have misaligned beyond recovery. The final phase is the sliding phase, in which the parts that have the same rotational alignment slide into lateral agreement.

This phase structure results because of the assumptions on the condition of the robots: namely, the connectors have an arbitrarily small inertia compared to the mass of the robot, there is no friction, critical damping and no bouncing.

In all cases, we exclude initial conditions where the faces do not come close enough to mate. When the forcing direction is vertical, this is equivalent to $x_0 > 2D * r_p + H$. We call this the "distance condition", and it applies to all connectors.

3.1.1 2D Kinematic Analysis: X-Face

We start by analyzing the conditions on the 2D X-Face. The approach phase can be handled simply by applying *a priori* collision detection. This gives us the point of first contact.

The alignment phase is next. When the first point contact is made, generically we will assume a vertex on one side contacts a line segment on the other side. The next step is to determine how the parts will rotate; towards alignment or away from alignment. We model this by building a simplified system seen in Fig. 3.3.

The direction of relative rotation can be determined by the geometry; the location of each rotation point (where the robot arm is applying translational forces but no torques) call them P_M and P_B, the location of the point of contact P_C and the normal to the line segment at the point of contact. Connector inertia $I_1 = I_2$ is arbitrarily small but nonzero.

We are modelling the system without friction, and so the only forces on both bodies at P_C must be in the direction normal to the line at P_C and in equal and opposite directions and the reaction forces at the rotation point (P_M or P_B). Since the moving face must rotate about P_M the force at P_C will cause a torque about P_M which will indicate its rotation direction. Similar analysis can be used for the base face. In addition to this torque, a slip occurs at P_C. The complication comes from the fact that both the contact slope and the location of P_C changes as motion occurs and in some cases the motion can reverse.

We numerically integrate this system to simulate the alignment phase. We only care about the path and not the rate, so the actual magnitudes of I and F are irrelevant and are excluded by setting them to 1.

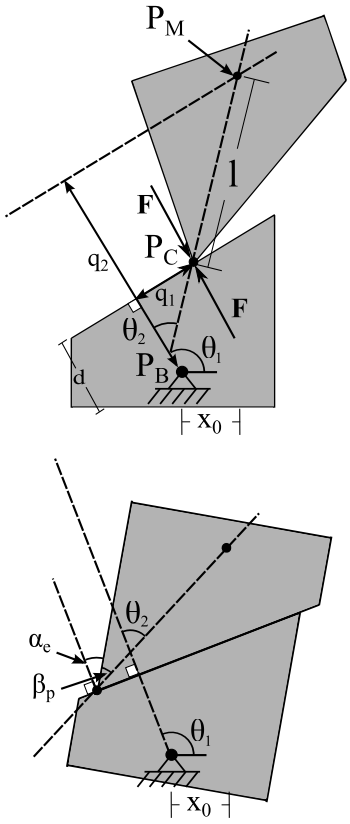


Figure 3.3: Alignment path

$$\begin{aligned} q_2 &= d + l\cos(\theta_2) \\ \ddot{\theta_1} &= -q_1 \\ (\ddot{\theta_1} - \ddot{\theta_2}) &= -\sin(\theta_2)l \end{aligned} \tag{3.1}$$

Together with the forcing condition $x = x_0$, this defines a one-DOF system.

The final equation we need is a motion constraint on the rotation point of the point side relative to the edge side. This is valid under the earlier assumption of arbitrarily large robot mass. We use the equation directly from the *Mating Problem Conditions*, and transform it to the new coordinate system $\mathcal{Q}$:

$$x = x_0$$

$$x = d\cos(\theta_1) + q_1\sin(\theta_1) + l\cos(\theta_1 - \theta_2)$$

Solving for θ_2, we get the differential equations:

$$\frac{I}{f}\ddot{\theta_2} = \sin(\theta_2)l - \frac{x_0 - d\cos(\theta_1) - l\cos(\theta_1 - \theta_2)}{\sin(\theta_1)}$$
$$\frac{I}{f}\ddot{\theta_1} = -\frac{x_0 - d\cos(\theta_1) - l\cos(\theta_1 - \theta_2)}{\sin(\theta_1)}$$

Rather than attempt to solve this system of ODEs analytically, we adopt a numerical solution here, timestepping through the system until the alignment condition is satisfied:

$$\frac{I}{f}\ddot{\theta}_{1i+1} = -\frac{x_0 - d\cos((\theta_1)_i) - l\cos((\theta_1)_i - (\theta_2)_i)}{\sin((\theta_1)_i)}$$
$$\frac{I}{f}\ddot{\theta}_{2i+1} = \sin((\theta_2)_i)l - \frac{x_0 - d\cos((\theta_1)_i) - l\cos((\theta_1)_i - (\theta_2)_i)}{\sin((\theta_1)_i)}$$
$$(\dot{\theta_1})_{i+1} = (\dot{\theta_1})_i + dt\ddot{\theta}_{1i+1}$$
$$(\dot{\theta_2})_{i+1} = (\dot{\theta_2})_i + dt\ddot{\theta}_{2i+1}$$
$$(\theta_1)_{i+1} = (\theta_1)_i + dt\dot{\theta}_{1i+1}$$
$$(\theta_2)_{i+1} = (\theta_2)_i + dt\dot{\theta}_{2i+1}$$

The alignment condition is:

$$\theta_2 = \beta_p + \alpha_e \tag{3.2}$$

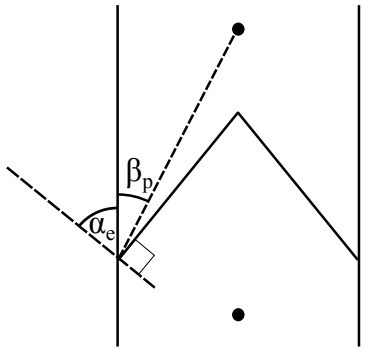


Figure 3.4: Angles for aligned faces. The V-Face is shown as an example.

where β_p and α_e are constant angles determined by the face geometry as shown in Fig. 3.3. β_p is the angle between the point face lever arm and its vertical. α_e is the angle between the edge normal and the edge face vertical. Misalignment conditions occur if $\theta_2 < 0$. This represents a *failed mate*, which we will talk more about shortly.

Once the faces are aligned, we move into the sliding phase. During the sliding phase, the two pieces are aligned (that is, $\theta_{moving} = \theta_{fixed}$), but there remains some (x,y) offset. Since we have excluded those situations which do not meet the "distance condition", we know that these two faces will at some point be close enough to dock. We also know that the derivative of the distance is negative, so the faces can only get closer. Aligned faces represent a stable equilibrium, so the only way for them to get closer is by sliding into the mating position. Thus, if we satisfy the alignment condition and the distance condition, we have a successful mate.

Likewise, a pair of faces which has failed to meet the alignment condition must necessarily be a failure. Even if it gets close enough, we will only get a "failed mate", which the system will not be able to recover from if the forcing condition is a constant.

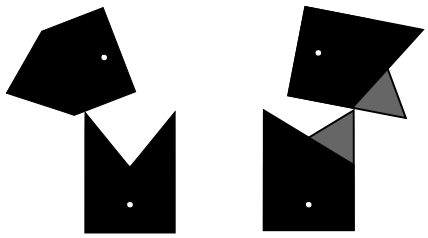


Figure 3.5: Examples of failed mates. Failed mates are unrecoverable for constant forcing conditions.

3.1.2 V-Face and S-Face

The earlier collision model is fine when only one-point and sliding contact cases are possible, as in the case of the X-Face. The S-face and V-face are slightly more complicated; there exists for these models a two-point contact case, as seen in Fig. 3.6. Rather than attempt to solve these cases, we establish bounds for these faces beyond which they will certainly fail. These characteristics give us an outer bound or "best case" on area of acceptance to compare to the X-Face simulation.

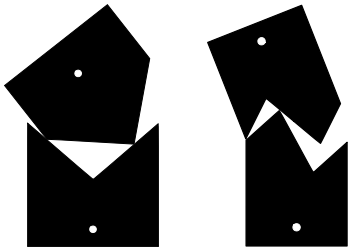


Figure 3.6: Example of two-point contact case.

The "best case" bounds for these faces are defined as follows. The path of the maximal point (e.g. the tip) of the moving face must contact one of the faces on the mating shape which would be adjacent to it in the mating minimal point. For the V-Face, this is mathematically equivalent to:$|x_{tip}| < D/2$. The other condition is that the initial angular offset does not already break the misalignment condition;

meaning $|\theta_0| < \frac{pi}{2} - |\alpha_e|$. Additionally, the V-Face and S-Face are subject to the same "distance condition" as the X-Face.

3.1.3 Simulation Results and Area of Acceptance Comparison

We ran the numerical simulator on the X-, S-, and V-Faces over a range of parameters. The relevant parameters here are $\frac{H}{D}$ and r_p. We tested each face over the same subset of the feasible set of initial states $\{-D < x_0 < D, -\frac{pi}{2} < \theta_0 < \frac{pi}{2}\}$. The subset of initial conditions is a combination of 41 x_0 values and 41 θ_0 values, giving us 1681 discrete points evenly spaced over both dimensions of the feasible area. Running the simulation on each of these points gives us a set of initial conditions paired with their successful/failed state. We use the number of these values which are successful as a metric M for that particular connector.

These metrics were generated for several values of the design parameters $\frac{H}{D}$ and r_p. M serves as a scalar discretized approximation of the area of acceptance we can use for comparison. We present the these metrics and their ratios in the Tables 3.2-3.4.

For the X-Face, the simulator was run once for each layer, comparing distances to see which layer collides first.

We can see that the X-Face generates a larger area of acceptance for most values in the design space. Some parameter values give better results with the "best case" of V or S-Face, but only in a specific narrow range. If we take the mean across the design space, we find that the mean X-Face/V-Face ratio is 1.8838, and the mean X-Face/S-Face ratio is 2.3846. In almost every case of $\frac{H}{D}$ and r_p the X-Face has significant improvement over both S-Face and V-Face area of acceptance. The few exceptions are near $r_p = 0$ and $\frac{H}{D} = 1$.

Looking at the data, we see some unintuitive results. As the rotation point comes closer to the face, the area of acceptance increases as we expect, however it drops off

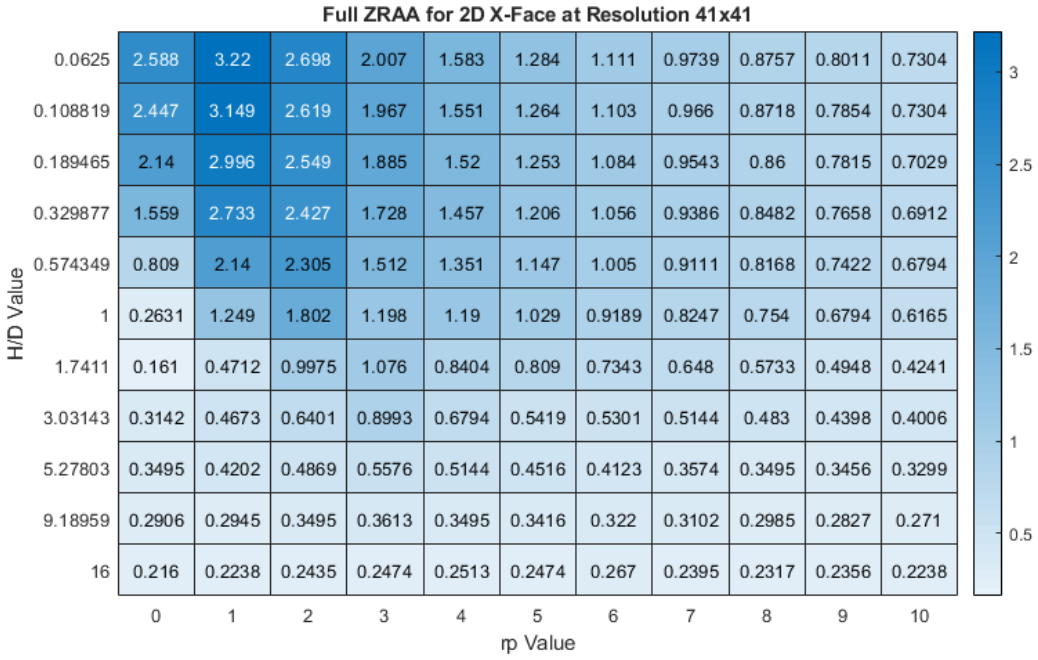


Table 3.2: Full ZRAA for X-Face by Direct Simulation

for $r_p = 0$. This is because of the distance condition; Once the rotation point is that close, significant offsets will result in failed mates simply because the faces do not come close enough together. As the $\frac{H}{D}$ ratio increases, the area of acceptance across all faces decreases. As mentioned for the "best case", there is some angular offset limit beyond which docking cannot succeed. Increasing the slope $\frac{H}{D}$, reduces this limit, naturally depressing the area of acceptance. More extreme values of $\frac{H}{D}$ and r_p were examined, but the area of acceptance "plateaus" and ceases to be interesting.

Each design has 1681 (41^2) initial conditions that are checked for acceptance, for 100 different combinations of design parameters for each face type. Additionally, some conditions (no friction, coef. of restitution) are impractical to replicate in the lab.

Simple experimental validation was performed by constructing small models of the faces and manually driving them together. Offsets (translational, and angular measured by tangent) are measured by use of grid paper beneath the models. Fig. 3.7

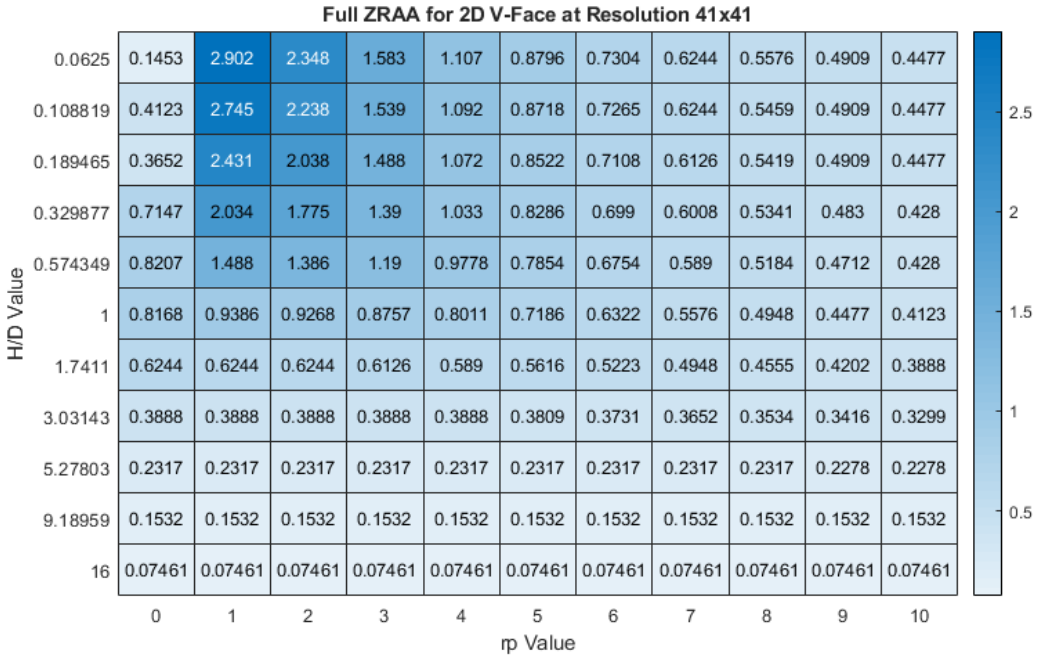


Table 3.3: Full ZRAA for V-Face by Direct Simulation

shows a typical run. The validation generally agreed with the model, except in cases where static friction became significant. In these cases the models were observed to "stick" and not move along the prescribed path. To find M for the equivalent real-world case models, we would need to add friction to our model. In general however, this serves as a useful point of comparison for connector geometries.

The X-Face and the analysis as presented thus far are limited in some regards. For example, the possibility exists in the real world case for some misalignment in the out-of-plane direction. A small design change can fix this misalignment. Taller layers and a gap between layers adds some tolerance to out-of-plane misalignment by keeping the appropriate layers in contact. This prevents small misalignments from causing complete failure to engage the connectors correctly. However, the model remains sound as a method of comparing these simple 2D connectors.

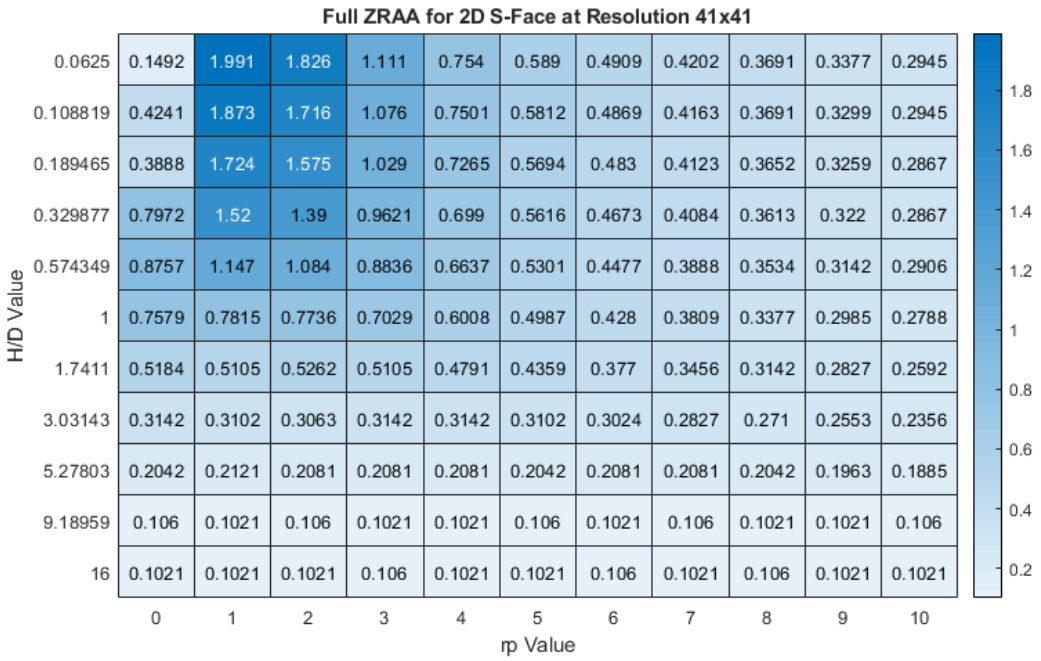
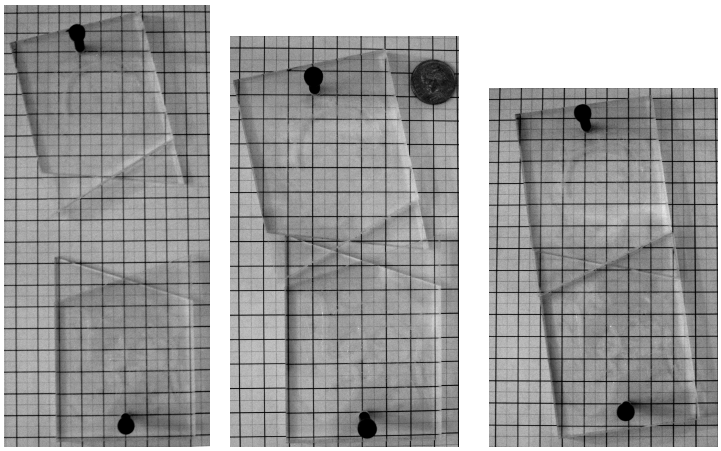


Table 3.4: Full ZRAA for S-Face by Direct Simulation



(a) Before alignment (b) During alignment (c) After alignment

Figure 3.7: Example of experimental validation procedure for X-Face with x_0=2cm, $\theta_0 = arctan(\frac{1}{3}) \simeq 0.32$. The grid pattern behind the transparent pieces has a 0.5cm spacing.

Chapter 4

3D Connector Alignment and X-Face

With the 2D connectors analyzed, we seek to move on to 3D alignment geometries. These are more useful for modular robotic applications in a 3D workspace. We start by describing existing 3D alignment geometries and a novel 3D connector design before working to compare them. We are able to show that the novel connector is a significant improvement over existing designs according to the metrics from Chapter 2.

4.1 3D Connector Designs

4.1.1 Existing Designs

Existing 3D alignment faces are common but generally limited in terms of capabilities - the most capable examples are the amour robot [68] and the DRAGON connector [49]. The ZRAAs are typically quite low as can be seen in Table 2.1, and so far have peaked at 1.57 for the amour connector. Their ZRAA comparisons can also be seen in Fig. 2.1.

4.1.2 3D X-Face Designs

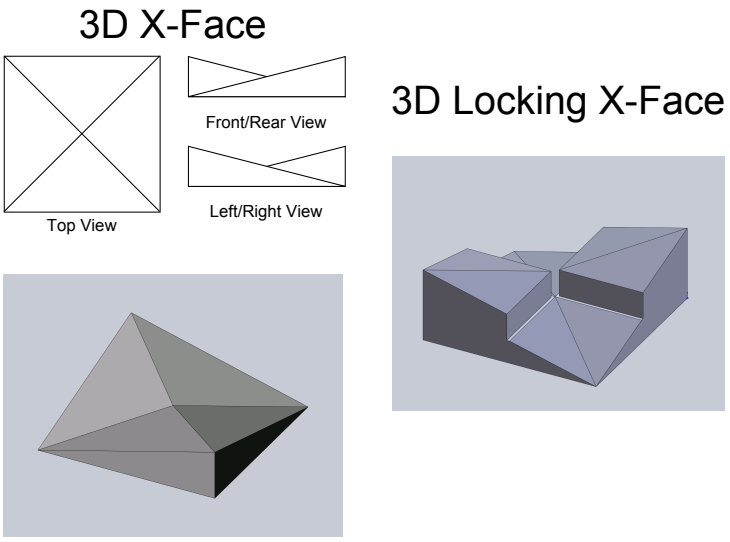


Figure 4.1: 3D X-Face Designs. A single connector is shown for each design- the full connector pair is 2 identical connectors.

Expanding on the design advantages of the 2D X-Face Connector we have designed the 3D X-Face connector. If we think of the design as a function over a 2D plane, it appears as a function with a 'saddle' point, two minima, and two maxima. The basic geometry of the 3D X-Face connector can be seen in Figure 4.1. We do not have the advantage we had on the original X-Face design of an extra dimension to be utilized, but by extending the aligning faces along the full length of the sides we still improve the effective AA of the connector, as we will show. This design has the advantage of being ungendered (that is, the docking faces have identical geometry), which helps with the operational flexibility of any resulting modular system by making every site eligible for docking. The design is also capable of correcting for some offsets in the *yaw* direction; that is, the rotational DOF about the direction of facing. The 3D X-Face connector has a ZRAA of exactly 2, a 27% increase over the amour connector, the next largest. For the entirety of this section, we use the X-Face geometry which has an angle of $\frac{\pi}{6}$, that is, $\frac{H}{D} = \tan\frac{\pi}{6} = 0.57735$.

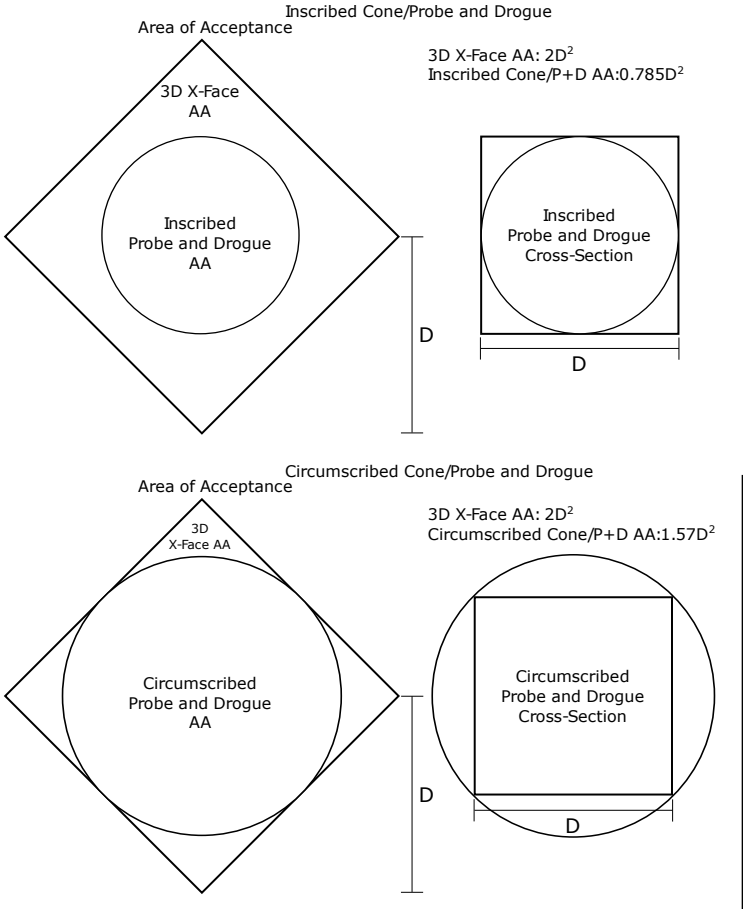


Figure 4.2: Comparison of ZRAAs for Cone/Probe-and-Drogue vs 3D X-Face connectors

The ZRAA Sum of a 3D face is usually readily apparent based on the maxima and minima. Comparison of the 3D X-Face to the Probe and Drogue shape is somewhat difficult because the cross sections have different geometries - the X-Face is square while the Probe and Drogue is a circle. We can choose either to circumscribe the X-Face with the Probe and Drogue cross-section or inscribe it. The circumscribed Probe and Drogue has a ZRAA of 1.57 compared to the 3D X-Face ZRAA of 2.0 - this represents a 27% increase of the X-Face over the the best existing gendered connector. Previously, the ungendered connector with the greatest ZRAA Sum was the DRAGON connector with 0.353, which the 3D X-Face improves on by 467%. Using the Oriented N-Cube Metric, the ZRAA is measured as 1.0, equivalent to the Circumscribed Probe and Drogue. The DRAGON measured using the same metric is 0.474, so the X-Face 3D still represents an improvement of 111% for ungendered connectors in this metric.

Another is the Locking X-Face geometry, which is a variant on the X-Face. The Locking X-Face is more robust to disturbance once connected since the addition of a lip feature at the saddle point prevents the connectors from coming apart in any direction other than the facing direction. Specifically, this lip feature is the addition of a vertical (z-oriented) mating face at the center lines along each edge, meeting at the saddle point. Theoretically any geometry can add a lip feature such as this one without affecting offsets in that plane (x,y,θ).

Twisting forces about z (or shear forces in x and y) on the 3D X-Face without the locking feature causes the two faces to separate in the z direction. The speed of this separation is linear with the tangent of the normal of an 3D X-face surface to z-axis (note that all four faces have the same value). Rotations about the z-axis yield a separation motion in proportion to the self-alignment behavior. When the surface normals are perpendicular to the z-axis, the tangent is infinite, and so are the separation velocities. This also corresponds to requiring infinite twisting force (up to material strength) to achieve separation. The Locking 3D X-face exploits this

principle by adding vertical faces in a lip feature, preventing twisting separation.

Unfortunately the lip feature introduces the possibility of jamming due to *pitch* or *roll* misalignments or foreign material. To combat the disadvantage, the mating face of this lip feature can either be given a 'draft angle' as one would in mold making. Adding a draft angle reduces the locking force from infinite, and can help prevent total failure due to dirt or other foreign object and reduce the possibility of jamming.

From a design standpoint, the locking feature also means we only have to add features to limit the approach DOF if we wish to ensure rigid connection once the connectors are docked successfully.

4.2 Analysis by Simulation

The area of acceptance for the 3D X-Face was first analyzed by direct simulation; that is, beginning a simulation with some offset and allowing it to run until the connector is either fully rejected or accepted. The simulation is detailed in the following sections. This setup uses the problem specification in Table 2.3 but only samples certain two-dimensional axes 'slices' for analysis due to the heavy computational cost.

4.2.1 Simulation Environment

Since we are now working in $SE(3)$, the Full AA has five dimensions . This increase in dimensionality not only makes it more difficult to analytically determine the AA as we did in Section 3.1, but also increases the cost factor of numerical computation. Beginning our work in this simulator with the ZRAA, we found that two-dimensional **Slice** areas of acceptance for unconstrained connectors were testable at a reasonable resolution within 12-18 hours on a standard laptop. We utilize the Gazebo dynamic simulation environment to evaluate and verify the X-Face connector's AA. We chose

Gazebo because it is relatively easy to set up and automatically repeat a simulation, and because it is capable of changing physics parameters relatively easily. This simulation environment is also highly capable of implementing friction, coefficient of restitution effects, multi-body systems, etc., making the simulation expandable to more of the dynamic and design space in which these connection problems occur. Gazebo also has multiple tools that can we used to ensure that the simulation's dynamic parameters (i.e. center of mass, inertia matrix) are correctly set and visible directly in the simulation. The default ODE physics engine was used.

Parameters, models and environment are determined by creating a standard Gazebo world file. A custom C++ plugin for Gazebo logs the results from repeating the simulation with different offsets. Alignment or failure to align was determined by measuring the distance between the 'saddle points' of the two X-Face connector models. If the two connectors are perfectly aligned, these two points are the same. In simulation this distance was checked every few frames. If the distance became less than a small threshold the parts were recorded as 'aligned', but if they become too far apart (more than the width of the connector) or if the simulation runs unusually long, the parts were recorded as 'failed to align'. Models were constructed in SolidWorks and exported in STL format then imported to Gazebo to form the visual and collision models for the 3D X-Face. To ease computation, the simulation was made essentially quasi-static with high forces and large global damping. Friction and coefficient of restitution are set to zero. While this condition is not one that occurs realistically, we are attempting to evaluate the alignment geometry, without choosing a material or robot. Thus it would represent an improper effect on the results to make any assumptions about the dynamics or material characteristics of the system. Additionally, we seek to isolate only the most basic mechanical interactions of the geometries in the scope of this experiment. This quasi-static condition is therefore the most system-neutral. The test performed to evaluate AA for the connectors in

each case is a simple 'drop test'. The parts are placed over one another with initial offset and dropped under gravitational force. One part rests on a ground plane so that it does not fall away from the other part under gravity. Center of mass is placed at the connector's center of mass, so gravity forces act through that point. The constant gravity force imposed in simulation is equivalent to an arm with force control on the end effector in that same direction, demonstrating a real-world case in which these results are directly applicable.

Numerical Limitations

While checking the results of the tests, regions were occasionally observed to have acceptance properties that are incorrect according to the properties of areas of acceptance (ie. asymmetric where they should be symmetric, holes or disconnected regions of acceptance. However, this was rare ($\leq E$ where $E \approx 0.08\%$). These errors were determined to be due to a combination of systematic floating point rounding error in the physics engine, and an edge case that caused collision checking to fail catastrophically. This effect appears more often near the boundary of the AA due to the more extreme angles involved. One set of initial conditions near the boundary was simulated $\approx 12,000$ times with 9 failures, while a set of initial conditions near the absolute center of the AA resulted in 0 failures after over 100,000 trials. Since the simulations sometimes consisted of upwards of 5,000 trials (ie. X-Pitch Slice with $71^2 = 5041$ trials), occasionally this problem resulted in a disruption of the results. To combat this, we reimplemented the simulation plugin to perform each trial 3 times, with the majority result being accepted. This reduces the probability that a given data point is incorrect to $\approx 1.92 \times 10^{-6}$, but triples the computation time required.

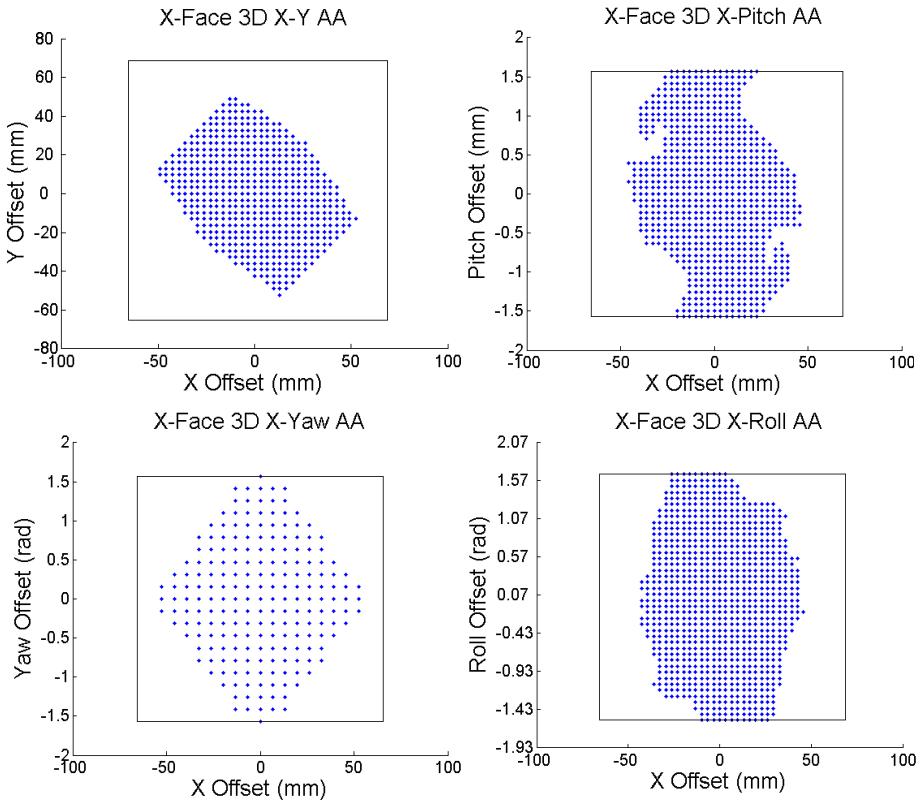


Figure 4.3: Results of Simulation Testing across four selected Areas of Acceptance. Inner rectangles represent feasible range of acceptance conditions, which is the range over which acceptance was tested.

4.2.2 Simulation Results

A mechanically simplified version of the simulation verified the ZRAA. This was performed in Gazebo as described above. As a way to prevent rotation, the mass properties were altered. By increasing the inertia five orders of magnitude, position changes took extreme precedence over rotational changes. The numerical simulation results are presented side-by-side with the analytical diagram in Figure 4.4. The results show a nearly perfect match between the simulation and the expected results. This verifies both that our simulator is capable of reproducing the ideal situation and that the ZRAA results are as expected.

The X-Y Slice, X-Pitch Slice, X-Roll Slice, and X-Yaw Slice AA an be seen in

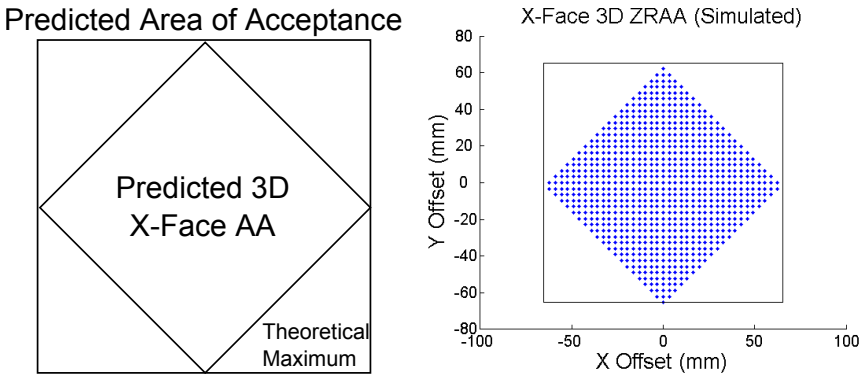


Figure 4.4: ZRAA Comparison. Left: Theoretical ZRAA based on geometry. Right: Simulated ZRAA using Gazebo

Figure 4.3. For these AA the inertia was returned to the correct form, but dynamic effects were reduced. To that end the gravity force was amplified (g=9000m/s^2) as well as adding a high degree of damping. To accomplish the damping, we set the exponential velocity decay term α for both angular and linear velocity to 0.99. While

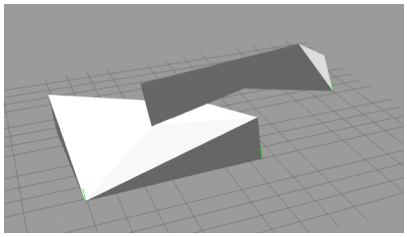


Figure 4.5: A stable configuration possible when rotations are permitted. Both connectors are in contact with the ground plane.

the ZRAA simulation results are straight forward, the other areas of acceptance have a more irregular shape. First, because we allow rotations in these cases, the position range is reduced, as in the X-Y Slice AA. Second, unaligned but stable configurations arise which are the cause of the concavities visible in the other AAs. An example of a stable configuration can be seen in Figure 4.5. The geometry of the faces allows for a considerable range of angular offsets to be successful. Even when rotated $\frac{\pi}{2}$ in *pitch* or *roll*, the shape of the edges in conjunction with the location of the center of

mass causes forcing conditions to be favorable for the connector.

Results show a relatively high acceptance for the 3D X-Face, taking up a large section of the available space. The Sum metric measurements from the simulation data for X-Y Slice, X-Roll Slice, X-Pitch Slice, and X-Yaw Slice are 0.336, 0.527, 0.481, and 0.504, respectively[1]. It is difficult to accurately predict the full-dimensional Sum value from these values, though we will show other methods in Chapter 5 that allow us to get closer to the Full Sum.

We also seek to estimate the Full Oriented N-Cube Metric introduced in Section 2.2.2. An optimistic estimate is the largest n-cube that can be formed that is consistent with the given two dimensional slices. This is found by the maximum circumscribed cube about the minimum of maximum inscribed squares of each slice. For orthogonal slices, this reduces to finding the n-cube as the one with a side length equal to the smallest of the Oriented N-Cube Metrics for each slice. The Oriented N-Cube Metric side lengths for X-Y Slice, X-Roll Slice, X-Pitch Slice, and X-Yaw Slice are 0.390, 0.536, 0.415, and 0.524 respectively. So the optimistic Oriented N-Cube metric is estimated to be 0.390. We can also use this as an optimistic estimate for the Full Sum AA.

A conservative *estimate* of the Full Sum can be made by assuming a 'diagonal' linear relationship between 2D planes; this results in a sort of five-dimensional 'diamond' shape. We solve this by creating a hyperplane in the positive orthant with the corner points from each of the four squares we have solved for in simulation (plus an extra point relating y and yaw; symmetry means X-Yaw Slice is identical to Y-Yaw Slice). We then solve for the point on that hyperplane that crosses the vector from the origin along (1,1,1,1,1). This gives us a vector (d,d,d,d,d) where d is the size of our n-cube. This can best be visualized as the cube centered at the origin that

[1]Note these slice AAs are on a 0-1 scale rather than scaling with π as in many later cases where rotation is present. In general the maximum value depends on the number of rotation DOFs ie. 0-π for 1 rotation DOF,0-π^2 for 2 rotation DOFs, etc.

fits within this 'diamond' shape. For the simulation data, we found the conservative value to be 0.204. This means the actual Full Sum Metric is likely between 0.204 and 0.390.

4.2.3 Prototype Construction and Testing for ZRAA

To verify our simulation results and make use of this design to dock our modular robots, prototypes of the 3D X-Face were constructed. These prototypes were constructed on the 65.5 mm x 65.5 mm scale, corresponding to the face size of the present version of CKBots [16]. The prototypes were 3D-printed on an Objet30 Photopolymer Printer out of VeroBlack material. This 3D-printing process has the advantage of giving a smoother finish than fused-deposition modeling. The lack of friction in our prototypes aids in matching the conditions in the simulation.

The full testing setup can be seen in Figure 4.6. An overhead arm composed of CKBot modules carries out a a simple up-and-down vertical trajectory while the cart on casters on the bottom is free to move in the plane for alignment. The arm trajectory moves the top connector up to a height that allows the connectors to clear each other and slide apart for different positions, and then down to the height at which fully aligned contact normally occurs. The face with the top connector does not rotate during this trajectory and maintains a level orientation. Grid paper below the cart and a positioning jig let us change and measure the offset by hand to an estimated error range of $\pm 1mm$. This test serves to experimentally verify the ZRAA to within our error range.

Testing was carried out at select points on the boundary, in 0.5cm increments as seen in Figure 4.7. All points inside the expected boundary were observed to align successfully, despite the degree of proximity to the boundary. Several points were tested that were presumably 'on' the boundary - these points aligned successfully as well. Points outside the expected boundary all failed to align successfully - the bottom connector was 'rejected' - pushed away. This testing confirms that the ZRAA

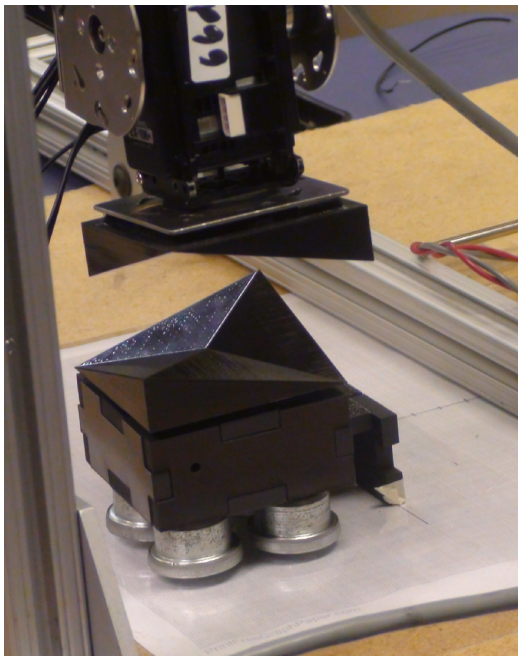


Figure 4.6: Alignment test setup. Overhead arm composed of CKBots. Jig aids in setting exact position.

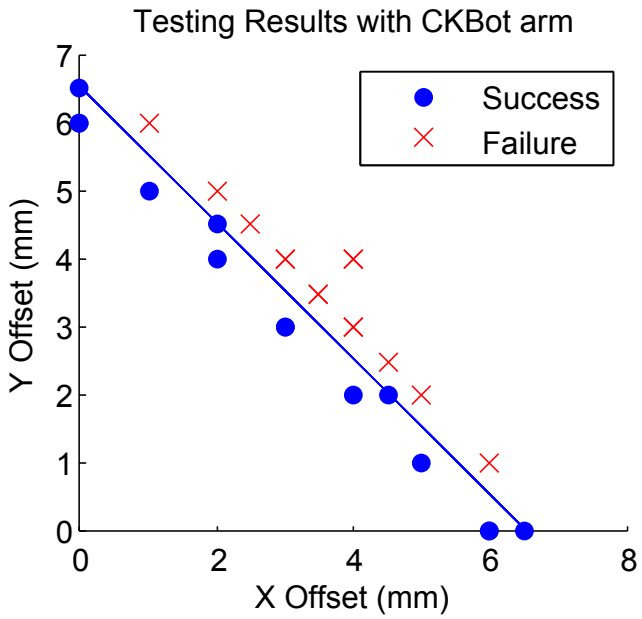


Figure 4.7: Results from testing on CKBot arm platform in Fig. 4.6. Natural symmetry of the geometry means testing requires only a single quadrant without loss of accuracy.

performs as expected on a real platform.

This testing was also able to demonstrate *yaw* correction for a few simple cases, including mixed correction of *yaw* with x and y offsets. Fig. 4.8 shows an example of the correction behavior.

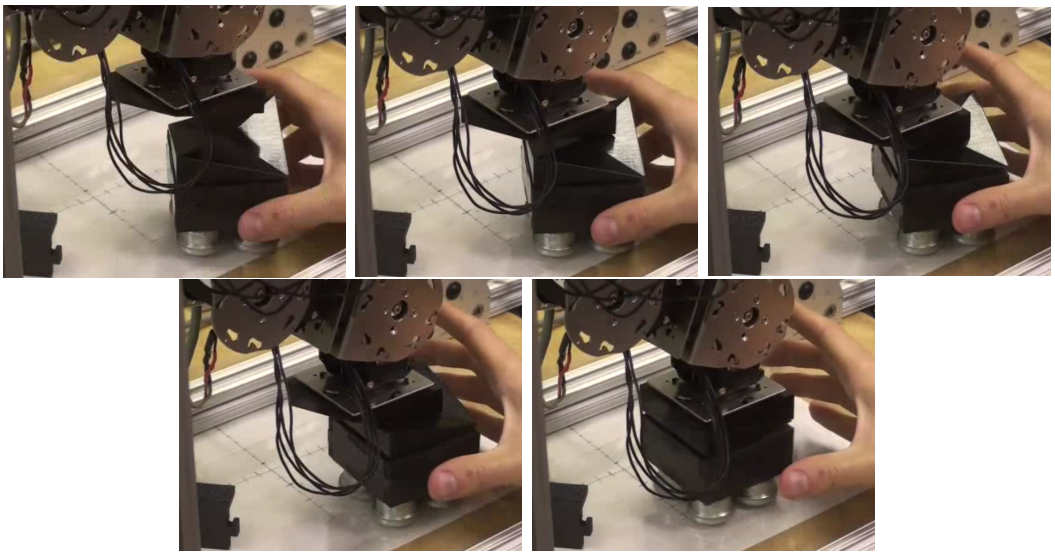


Figure 4.8: Motion sequence showing alignment correction of offsets in x, y, and *yaw* simultaneously. Yaw rotation is the first major motion visible, followed by translation in x and y simultaneously.

Chapter 5

Numerical Methods for Determining Area of Acceptance

While the simulation method gives passable results and resulted in a measurement of area of acceptance within some level of statistical error, it would be better to have a more precise numerical analysis method to determine AA. We have found two (2) processes by which we can obtain a more consistent and precise result which is detailed in the following sections. These two methods share some foundational concepts but are in many ways very different in terms of implementation.

The same assumptions used in the simulation approach are necessary here, namely; zero friction, pseudostatic interactions, zero coefficient of restitution. The forces influencing the system are still purely mechanical interaction forces with no dynamic interactions. Since many robotic systems can be run almost arbitrarily slowly, this is a good starting point for robust acceptance; we assume that any given robotic system can be run sufficiently slowly to approximate the pseudostatic condition. Frictional interactions can in many cases be overcome by sufficient effort from the robot, so we also exclude the consideration of friction as a measure of simplification. Additionally, in practice there are many processes by which we can reduce surface friction - ie. lubricants, surface treatments. For the prototypes of the 3D X-Face (printed on

an Objet30 desktop 3d printer) choosing a polished surface finish from the printer is all that is necessary to reduce the friction to a negligible level for alignment on our testing platforms. Additionally, as in the previous case, all area of acceptance metrics are normalized by the width D of the connector. This ensures that we can compare geometries fairly - larger connectors (that is, those with a wider base D) would naturally have higher area of acceptance for the same geometry.

For the purposes of this chapter, we use some standard Alignment Problem Specifications as in Section 2.3. The 2D problems are the same as in Table 2.2, and an example 3D problem specification with restricted DOF can be found below in Table 5.1. Similar specifications apply to other problems i.e. (x,y,yaw) or (x,$roll$,$pitch$) when they are analysed.

Condition	State
Forcing Direction:	$z = z0$
Frictional:	Zero Friction
Damping:	Overdamped
Restitution:	Zero
Initial Offset Dimensions:	2 linear(x & y) and 1 angular ($pitch$)
Free Dimensions:	2 linear(x & y) and 1 angular ($pitch$)

Table 5.1: Alignment Problem Specification for X-Y-Yaw AA, a three-dimensional AA with only 3 free DOF

5.1 Generating Configuration Space Obstacles for Watershed Segmentation

These approaches use the concept of **configuration space** (C-space) to determine how the system might behave under contact forces. This concept originates from

motion planning, and is the space of all possible states for which the robot is not in collision with any obstacles. The robot can be shrunk to a point and the obstacles grown (C-obstacle) so that collision-free motion can be examined as a point moving through the space [40]. The resulting obstacle geometry contains on its boundary all possible contact states for the robot and obstacle. Methods for finding these obstacles are well-developed thanks to their uses in collision checking and path planning. Depending on the accuracy required, fully-dimensional configuration-space representations can be computationally costly. A survey of methods for constructing these obstacles can be found in [72]. Since the outer boundary of the configuration space obstacle represents the set of points in which the connectors are in contact, we can analyze the shape of this surface (with some assumptions) to find a presumptive area of acceptance for two rigid connectors.

Similar concepts have been presented in works on stability analysis and robotic grasping. Rimon and Burdick [43] analyze grasp mobility and stability in terms of the configuration space representation of the system. Contact forces are represented within the configuration space using screw theory. Further analysis characterises mobility of a grasp using 1st and 2nd order analysis to show lower bounds on the number of contacts required to effectively grasp an object [54] [55]. The outer boundary of the configuration space is known as the *contact space*, a useful manifold for determining grasp quality [39] or finding configurations to perform caging [2].

The **watershed** of a point in an image (or height map) is defined as the set of all points in the image with a downstream in which that point is a member [8]. In other words, the watershed of a point is the set of all points which flow downhill to it. The watershed of our target configuration is therefore equivalent to the AA where the downhill direction vector is defined to be the approach vector. Many methods exist for determining the watersheds of an image developed by the image segmentation community [56] [69]. The AA is also a specific case of the 'pre-image' of the target configuration as defined by Lozano-Perez, Mason, and Taylor [41].

Each of the two processes used to numerically determine AA requires a) generating some representation of the contact space, and then b) searching for the watershed on that contact space (which represents the AA).

For each method, the geometric functions have been implemented in Matlab with the use of the Multi-Parametric Toolbox [36], which enables computation and operations on sets of polytopes. Final analysis is implemented in Matlab in an original code base.

The first method has five steps, which are each detailed and discussed in Sections 5.1.1, 5.1.2, and 5.3.

- Define connector geometries
- Generate Surface Collision Shape
- Reduce to Sub-manifold of interest (Contact Space)
- Generate Poly-vector pairs
- Follow pairs to generate AA

5.1.1 Defining Connector Geometries

We define a **connector pair** as $(\mathcal{C}, \mathcal{C}')$ where each of $\mathcal{C}$ and $\mathcal{C}'$ are *P-collections*. A P-collection is defined as in [4]:

Definition: *A P-collection* $\mathcal{C}$ is a finite collection of full-dimensional polytopes in $\mathbb{R}^n$, i.e.,

$$\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^{N_C} \tag{5.1}$$

where $N_C = card(\mathcal{C}) < \infty$,$\mathcal{C}_i := \{x \in \mathbb{R}^n | \mathcal{C}_i x \leq c_i\}$,$\mathcal{C}_i$ full-dimensional, i=1,...,N_C. A *polytope* is defined as

Definition (Polyhedron and polytope): *A convex set*

$$\mathcal{P} = \{x \in \mathbb{R}^n | Px \leq p\}, \tag{5.2}$$

with $P \in \mathbb{R}^{n_p \times n}$,$p \in \mathbb{R}^{n_p}$,$n_p < \infty$, is called *polyhedron*. A bounded polyhedron is called a *polytope*. The vector in 5.2 is considered component-wise. A polytope $\mathcal{P}$ is *full-dimensional* if it is possible to fit a non-empty n-dimensional ball in $\mathcal{P}$, i.e.,

$$\exists x_0 \in \mathbb{R}^n, \epsilon > 0 : \mathcal{B}(x_0, \epsilon) \subset \mathcal{P}, \tag{5.3}$$

where

$$\mathcal{B}(x_0, \epsilon) := \{x \in \mathbb{R}^n | \|x = x_0\| \leq \epsilon\}. \tag{5.4}$$

For $\mathcal{C}$ and $\mathcal{C}'$, each polytope in each of the sets should be full-dimensional in either $\mathbb{R}^2$ or $\mathbb{R}^3$. The nth dimension in the space, either y in $\mathbb{R}^2$ or z in $\mathbb{R}^3$, is assumed to be the 'approach direction'; for the sake of generality I will refer to this dimension as e_p. Note here that each of $\mathcal{C}$ and $\mathcal{C}'$ should be given in the orientation with the mating faces pointed in the positive approach direction e_p, and not in an orientation where they would normally mate. This prevents any confusion when performing the set operations necessary to the rest of the process. Examples of appropriate orientations can be seen in Figures 5.1 and 5.2.

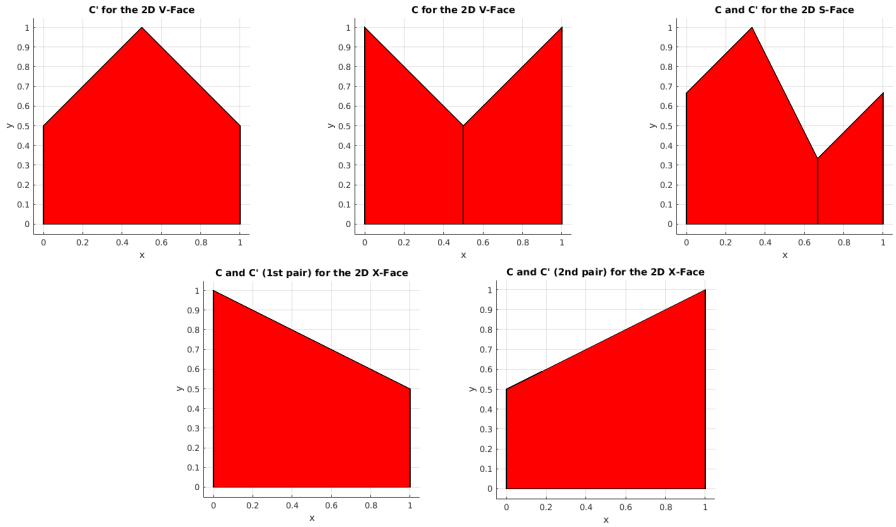


Figure 5.1: Two-dimensional connectors represented as P-collections $\mathcal{C}$ and $\mathcal{C}'$

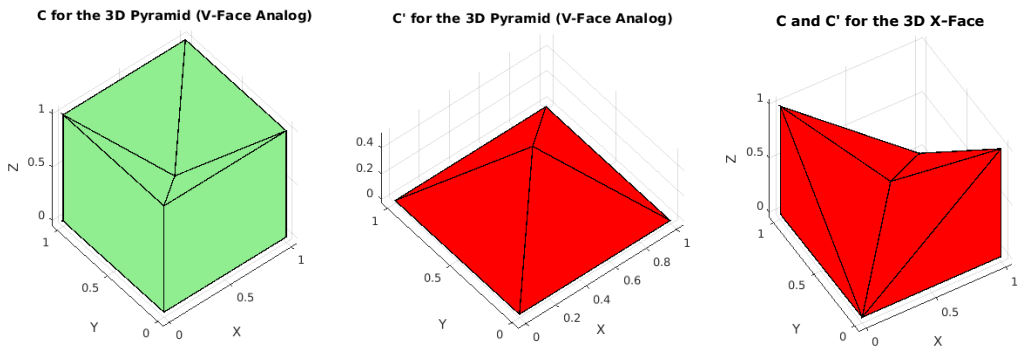


Figure 5.2: Three-dimensional connectors represented as P-collections $\mathcal{C}$ and $\mathcal{C}'$

If the underlying sets of $\mathcal{C}$ and $\mathcal{C}'$ are equal or there exists some offset x such that the two are equivalent, then we say that the connectors are **ungendered**. If no such equivalence exists, the connectors are **gendered**.

Definition (Ungendered Connectors): *A connector pair* $(\mathcal{C}, \mathcal{C}')$ *is said to be* **ungendered** *iff*, given $\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^{N_C}$, $\mathcal{C}' = \{\mathcal{C}'_i\}_{i=1}^{N'_C}$, $\exists x \in \mathbb{R}^n$

$$\exists x \in \mathbb{R}^n | \bigcup_{i=1}^{N_C} \mathcal{C}_i := \bigcup_{i=1}^{N'_C} \mathcal{C}'_i + \mathbf{x} \tag{5.5}$$

The addition operation above is the translation of the set, i.e ,$(\mathcal{P} + \mathbf{x}) = \{p + \mathbf{x} \in \mathbb{R}^n | p \in \mathcal{P}\}$.

5.1.2 Generating Surface Collision Shapes

In alignment we are interested in surface interactions. In order to obtain an idea of where the parts interact, I construct a surface collision shape for a given pair of connectors. This collision shape construction involves the Minkowski-Addition operation [36]:

Definition (Minkowski Addition): *The Minkowski-Addition of two polytopes* $\mathcal{P}$ *and* $\mathcal{W}$ *is a polytope*

$$\mathcal{P} \oplus \mathcal{W} := \{x + w \in \mathbb{R}^n | x \in \mathcal{P}, w \in \mathcal{W}\} \tag{5.6}$$

Specifically, we perform the operation $\mathcal{C} \oplus \mathcal{C}'$ to get the surface collision shapes. The Minkowski Addition operation constructs a shape representing relative positions that would be in collision. Any coordinate point in the resulting shape, if it were given as a relative position of the two connectors would result in an overlap. On the boundary of this shape the two connectors are just barely in contact. It is these cases where interaction forces would come in in the real world, so these are the ones we are interested in.

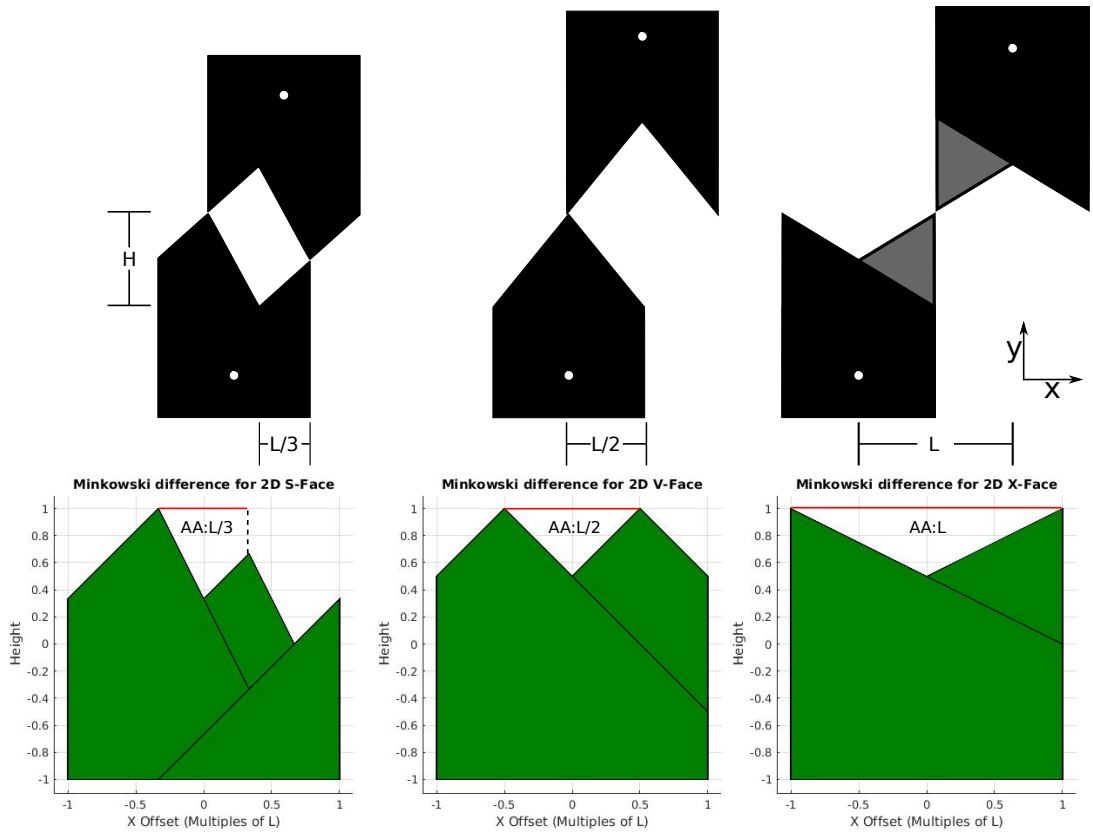


Figure 5.3: Surface Collision Shapes (P-collections) from Minkowski difference for strictly translational 2D cases, with typical connector types

To visualize why this works, we recast the system as an example system from path-planning. We let the 'robot' be $\mathcal{R} = -\mathcal{C}'$, and the 'obstacle' $\mathcal{B} = \mathcal{C}$. If we were to construct a configuration-space representation of this obstacle, we would perform the Minkowski *difference* (aka Pontryagin Difference) [36] on $\mathcal{B}$ and $\mathcal{R}$:

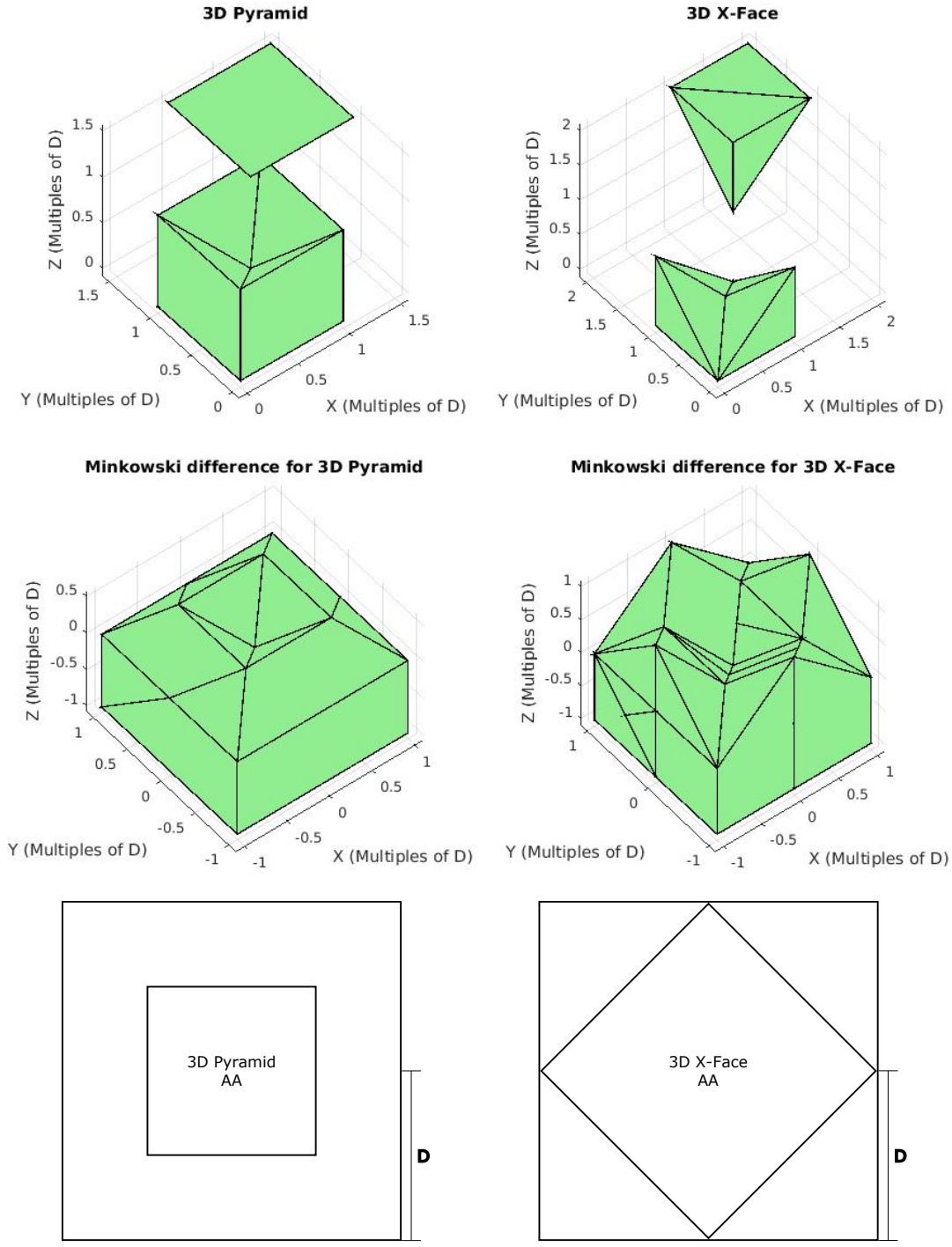


Figure 5.4: Surface Collision Shapes (P-collections) for strictly translational 3D cases, with typical connector types

Definition (Minkowski/Pontryagin Difference): *The Minkowski Difference of two polytopes $\mathcal{P}$ and $\mathcal{W}$ is a polytope*

$$\mathcal{P} \ominus \mathcal{W} := \{x \in \mathbb{R}^n | x + w \in \mathcal{P}, \forall w \in \mathcal{W}\} \tag{5.7}$$

It is true that if $\mathcal{P}$,$\mathcal{W}$ are subsets of some affine space (in this case, $\mathbb{R}^n$); so then $\mathcal{P} \ominus \mathcal{W} = \mathcal{P} \oplus (-\mathcal{W})$ [37], so we can see that $\mathcal{B} \ominus \mathcal{R} = \mathcal{C} \oplus -(-\mathcal{C}')$, meaning these problems are equivalent.

This Minkowski operation gives us a shape that represents the full set of *relative* coordinates for which the two connectors have a non-empty intersection, or in other words, are in collision.

Strictly Translational Areas of Acceptance

For a strictly translational areas of acceptance, the aforementioned $\mathcal{C} \oplus \mathcal{C}'$ operation in our numerical toolbox is sufficient to get the collision shape. A few of the collision shapes generated in this manner are shown in Figures 5.3 and 5.4 along with the corresponding connectors and areas of acceptance.

Note that the areas of acceptance correspond to a watershed connected to a local minima, specifically the one that contains the target mating configuration. The area of acceptance therefore is the region of attraction to the minima corresponding to the target location (that target location being the 'zero offset' or 'fully aligned' position).

Translational and Rotational Areas of Acceptance

When adding rotational offsets, the problem becomes more complicated. Not only does the problem dimensionality increase, but also the degrees added are continuous rather than discrete, so in practical situations we are usually *approximating* the exact collision shape with some level of accuracy.

In the two-dimensional case with rotations (offsets in x, y, and θ) we can approximate the collision shape by taking slices at a subset of values in θ and then

compositing them together. This can be implemented in one of two ways: by first finding the Minkowski sum of each slice and then compositing them together, or by generating a representation of one of the connectors that includes θ - $\mathcal{C}_{full}$ - and then performing a Minkowski sum. We take the second approach in order to make composition easier and ensure the preservation of the critical concavities.

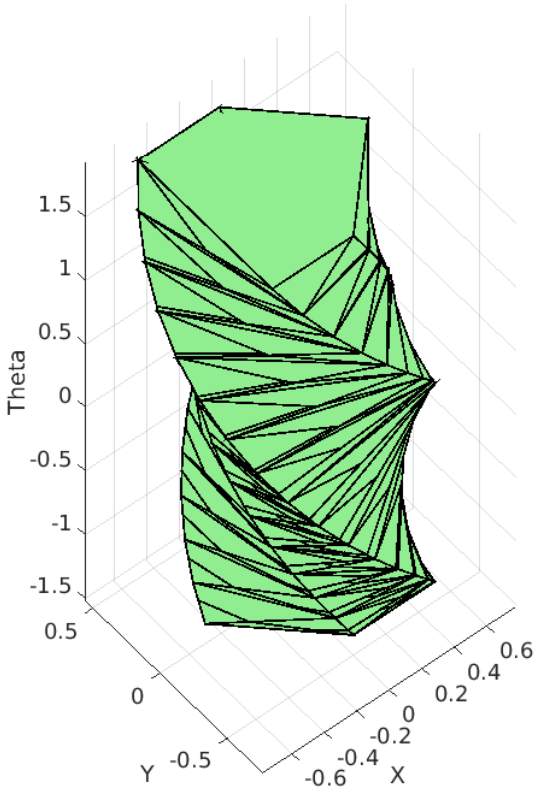


Figure 5.5: Rotation inclusive representation of $\mathcal{C}$, $\mathcal{C}_{full}$ for the 2D V-Face with center of rotation at the centroid of $\mathcal{C}$

To generate a representation of $\mathcal{C}$ that includes θ, we must first generate representations of the connector at different orientations θ. This is done by a simple rotation matrix operation through a rotation matrix R. Since rotation matrices rotate about the origin, we first translate the connector so that the connector is offset correctly the correct amount for the specified center of rotation to be at the origin. If the

center of rotation changes, this translation must be adjusted for the new center of rotation.

We then have several slices $\mathcal{C}_{\theta_i} = R(\theta_i) * \mathcal{C}, i = 1, 2, ..., n$, where n is the 'resolution' or number of slices. Each of these slices is only a representation of a single connector, rotated - we do not yet perform the Minkowski operation to bring the slice into configuration-space. We then composite these slices together into a fully-dimensional representation of the connector - e.g. one existing in $(x, y, \theta) \in \mathbb{R}^3$ Each slice is placed at a z position according to the corresponding θ, and then undergoes a composition process we call Corresponding Polytopal Composition(CPC) to obtain a full-dimensional P-collection, which I explain below. The full composite representation of the V-Face 2D connector $\mathcal{C}_{full}$ can be seen in Figure 5.5

This composite approximation to the full shape is created in such a way as to preserve concave features (approximate to the slice resolution), which is what we are interested in. To the best of our knowledge there exists no other method to perform this kind of rotational composition approximation operation on polytopes.

Given a desired resolution (number of slices), Corresponding Polytopal Composition combines two slices at a time, each time adding to a full-dimensional P-collection until the desired resolution has been reached. To do this we select a polytope from the P-collection of slice 1 and a vertex from the P-collection of slice 2. A polytope is generated from the set of this polytope and vertex, which is added to the new, full-dimensional P-collection. Since the polytope from slice 1 is convex and we are adding only a single out-of-plane point, the new polytope is also convex. This process is repeated for all combinations of polytopes in slice 1/vertices in slice 2, and then for vertices in slice 1/polytopes in slice 2. The full process is repeated for all slices. This results in a P-collection $\mathcal{C}_{full}$ in the full dimension of interest that represent an approximation to the interstitial volume for the slices of the connector orientation. This method can be used on higher dimensional P-collections with no real change to the basic method - to combine two slices we only need combine the polytope/vertex

sets from the slices in the same manner.

function combine slices $(\mathcal{C}, div)$

Input : P-collection $\mathcal{C} \in \mathbb{R}^2$, div $\in \mathbb{N}$

Output: P-collection $\mathcal{C}_{full} \in \mathbb{R}^3$

$\mathcal{C}_{full} \leftarrow$ Empty P-collection

for *Polyhedron* $\mathcal{P}_i \leftarrow \mathcal{C}_1$ **to** $\mathcal{C}_N$ **do**

  for $\theta \leftarrow -\frac{\pi}{2} : \frac{\pi}{div} : (\frac{\pi}{2} - \frac{\pi}{div})$ **do**

    $R(\theta) \leftarrow \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix}$;

    $R(\theta + \frac{\pi}{div}) \leftarrow \begin{pmatrix} cos(\theta+\frac{\pi}{div}) & -sin(\theta+\frac{\pi}{div}) \\ sin(\theta+\frac{\pi}{div}) & cos(\theta+\frac{\pi}{div}) \end{pmatrix}$;

    Slice $\mathcal{P}_i^{(\theta)} = R(\theta)\mathcal{P}_i$;

    Slice $\mathcal{P}_i^{(\theta+\frac{\pi}{div})} = R(\theta + \frac{\pi}{div})\mathcal{P}_i$;

    for $j \leftarrow$ *1* **to** $\nu_{\mathcal{P}_i}$ *(# of vertices in* $\mathcal{P}_i$*)* **do**

      // Polyhedron formed by all vertices in $\mathcal{P}_i^{(\theta)}$ and one from $\mathcal{P}_i^{(\theta+\frac{\pi}{div})}$

      $\mathcal{C}_{full}$ += Polyhedron$(\mathcal{P}_i^{(\theta)}, V^j_{\mathcal{P}_i^{(\theta+\frac{\pi}{div})}})$;

      // Polyhedron formed by all vertices in $\mathcal{P}_i^{(\theta+\frac{\pi}{div})}$ and one from $\mathcal{P}_i^{(\theta)}$

      $\mathcal{C}_{full}$ += Polyhedron$(\mathcal{P}_i^{(\theta+\frac{\pi}{div})}, V^j_{\mathcal{P}_i^{(\theta)}})$;

    end

  end

end

Algorithm 1: Corresponding Polytopal Composition - 2D connector case

Applied to multiple slices for the resolution of interest, we get the final $\mathcal{C}_{full}$. We then perform the Minkowski Sum $\mathcal{C}_{full} \oplus \mathcal{C}'_3$ to get the Surface Collision Shape, $\mathcal{SCS}$, which is also a P-collection. The Surface Collision Shape is an approximation in terms of *div* that represents the closed full-dimensional set of all points where the connectors would overlap, even if this is not physically possible.

Note that $\mathcal{C}'_3 = \{c \in \mathbb{R}^3 | (x, y) \in \mathcal{C}', z = 0\}$. In other words, $\mathcal{C}'_3$ is the original,

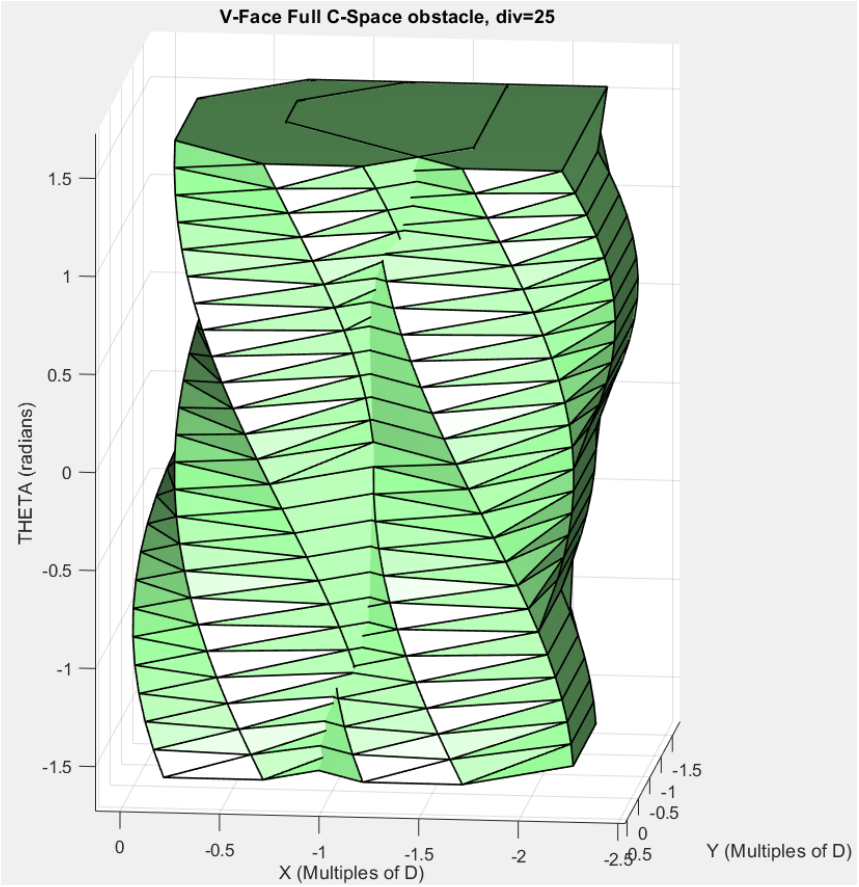


Figure 5.6: A surface collision shape for the V-Face Connectors given a resolution of 15 slices

lower-dimensional version of $\mathcal{C}'$ as a flat (zero-thickness) polytope at z=0 in the Minkowski sum, and not a full-dimensional or composite version. The z dimension represents rotation, so this prevents us from including excess rotation points that do not belong to the $\mathcal{SCS}$. Another way to understand this is to consider $\mathcal{C}_{full}$ as an approximation to a large stack of slices, each of which would be Minkowski-added to the $\mathcal{C}'$ to get the collision shape; the Minkowski operation can only be applied along the x-y plane in reality, so to do this in the approximation a flat representation of $\mathcal{C}'$ is necessary.

For a 3D connector, the process to obtain $\mathcal{C}_{full}$ is similar - composition of polytopes into a higher-dimensional P-collection that includes the possible rotations. The first difference is that we can specify different resolutions or slice numbers for each of the three rotational degrees of freedom $(\theta_{roll}, \theta_{pitch}, \theta_{yaw})$, although in general it is likely to bias the results unless the three resolutions are the same. The second difference is that we must repeat the process for each of these DOFs. So first we generate a 4-dimensional composite $\mathcal{C}_{full,\theta_r}$ by CPC. We then generate a five-dimensional composite $\mathcal{C}_{full,\theta_r\theta_p}$ from $\mathcal{C}_{full,\theta_r}$ by CPC again, then a six-dimensional composite $\mathcal{C}_{full,\theta_r\theta_p\theta_y}$ from $\mathcal{C}_{full,\theta_r\theta_p}$ by applying CPC a third and final time. The final shape $\mathcal{C}_{full,\theta_r\theta_p\theta_y} \in \mathbb{R}^6$ is Minkowski-added with the original $\mathcal{C}'$ in the same way as the 2D case to generate $\mathcal{SCS} \in \mathbb{R}^6$.

Aside from having a different representation for the rotation matrices corresponding to *roll*, *pitch*, or *yaw*, the 3D process for CPC (each of the three times) is the same.

5.1.3 Reduction to Contact Space

While this gives us an approximation of the full set of colliding configurations, we do not need all of these points to find the area of acceptance. Specifically, we only need the surface sub-manifold of the full Surface Collision Shape. We obtain this by taking the intersection of the Surface Collision Shape with the *closure* of its complement.

Finding the surface sub-manifold $\mathcal{S} \in \mathbb{R}^n$:

$$\mathcal{S} := \mathcal{SCS} \bigcap cl(\mathbb{R}^n \setminus \mathcal{SCS}) \tag{5.8}$$

In the implementation, the complement is found with the **regiondiff** operation [4] (a set subtraction of the Surface Collision Shape with the full space $\mathbb{R}^n$. This implementation automatically returns the closure rather than the open set. With these two closed sets intersecting only along the surface, the intersection operation gives us the surface sub-manifold as a union of convex polyhedra. This surface sub-manifold is the (approximated) *contact space*. This signifies that the space represents only those points where the objects are just touching. In real terms these are the possible offsets of the two objects that represent any contact, since larger overlaps between the two regions are physically unrealisable for rigid bodies and would be in the interior of the Collision Shape.

Furthermore for *our* analysis of the area of acceptance, we only require a single watershed. The watershed of the docking configuration represents the area of acceptance. We can reduce the region of interest then to only those with a height value in the approach direction e_p greater than that of the docking configuration :

$$\mathcal{S}^o = \mathcal{S} \bigcap \{x \in \mathbb{R}^n | e_p > 0\} \tag{5.9}$$

where e_p is y in the 2D case and z in the 3D case. This ensures two characteristics about the set of surface polytopes for the next set. First, there are fewer polytopes in the set, reducing computational complexity. Second, the polytopes which remain in the set correspond to faces with *positive outward normals* to the Surface Collision Shape. To acquire the outward normal first we take a cross product, then multiply by some number h to ensure the vector is positive in the approach direction.

5.2 Pre-Partitioning and Graph Traversal Method

The first method for obtaining the area of acceptance from the surface collision shape we call the 'Pre-partitioning' method [23].From this contact space sub-manifold $\mathcal{S}^o$ we next make use of a custom watershed algorithm implementation. The algorithm has two parts performed in sequence, which are described below. First however, we need to define a height function $h(\mathbf{x})$ on the space. This height function tells us in which direction to assume a point will lead. To visualize this, think of a drop of water placed on a surface. The drop will flow downwards along the surface based on gravity in the direction of steepest descent. The height function then tells us the 'height' of each point in the space, allowing us to find which points 'flow' towards the same minimums. The set of all points which 'flow' towards the same minimums are referred to as *catchment basins*, *basins of attraction*, or *watersheds*. In Figure 5.1, the height function is simply the z coordinate, so $h(\mathbf{x}) = z$. Note that in some figures ie. Figure 5.6 the height function is instead the y coordinate.

5.2.1 A Note on Watershed Algorithms

For the partitioning of a set of polyhedra into watersheds, research has focused mostly on partitioning of relatively high-resolution triangular meshes [42] [62] [66] by some variation of Meyer's flooding algorithm [7], using the curvature as the height function. This particular method does not suit the cases here; for one the data is not natively presented as a triangular mesh but rather as a polyhedral mesh, where each polyhedron may have more than 3 vertices. A triangular mesh could be obtained by triangularization of the existing mesh, but this could potentially be computationally less efficient than the polygonal representation.

The second reason is that there exist cases where the watersheds cannot be accurately represented by the set of available vertices, and thus a re-segmentation of the mesh is necessary. Figure 5.7 shows an example of a case where this is true; the

available set of vertices are insufficient to provide the correct watersheds. Even if the polygonal mesh is decomposed into a triangular mesh, there is no guarantee that the flooding algorithm will provide the correct watersheds. This happens because the traditional forms of the flooding algorithm are limited in their choice of partitions; the watershed border line must join some set of the vertices given in the polyhedra specification.

For large meshes with high vertex density this is acceptable; errors will only occur at the borders and be relatively small in area. However, the vertex density of many of the shapes generated by the CPC algorithm above are relatively low vertex density relative to the size of the area under investigation. Increasing the vertex density is possible by increasing the resolution in terms of number of slices, but this increases both the time required and the size of the data. In the case of 3D connectors, this is made significantly worse by the 3 dimensions of the resolution (*roll*, *pitch*, and *yaw*) which must all be increased, so the factor of increase is cubed.

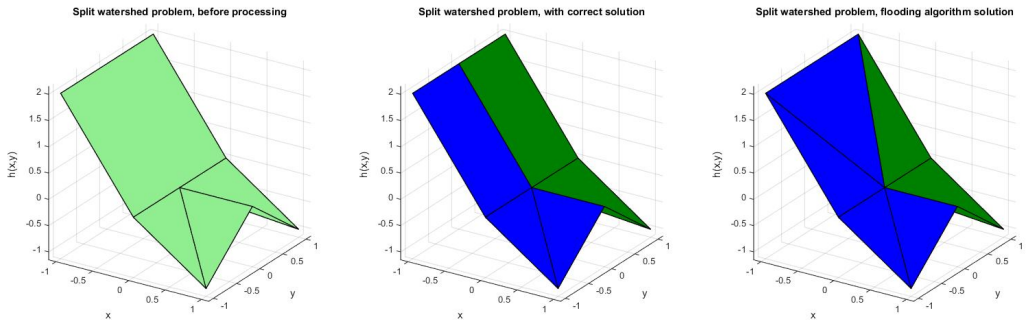


Figure 5.7: Split watershed problem demonstrating situation in which the watershed algorithm result is inaccurate

5.2.2 Pre-Partitioning Algorithm

The first step in the algorithm is proper pre-partitioning of the contact space. Each polytope should have all points in it 'flow' downhill towards no more than one other polytope. This means each polytope corresponds to a node in a directed graph,

specifically a *directed pseudoforest*. In a directed pseudoforest, all nodes have outdegree of at most one.

In order to ensure the polytopes are flowing towards one single polytope each, we first pre-partition the polytopes in the P-collection so that no polytope flows towards more than one other polytope. This ensures we get the correct solution to the split watershed problem in Figure 5.7. Each polyhedron is bounded and has a direction of steepest descent, and thus a (possibly empty) set of neighboring polyhedra to which points strictly descending along the polyhedra in this direction will go. To partition, we split polyhedra until each polyhedron has no more than one downhill polyhedron for all points inside it. This step can be computationally intensive, as it requires the determination of neighboring relationships for these polyhedra, a process that requires $O(N_p^2)$ time, where N_p is the number of polyhedra. The neighboring relationships must also be checked again for the new polyhedra after the split, though using only the subset of previous neighbors. This must also be run several times, as a polyhedron that is partitioned may result in the necessary partitioning of its uphill neighbors as well.

The procedure to partition the polyhedrons is as follows. First, we find a polyhedron's shared edges with neighbors by intersection. Next, we find the downhill edges by shifting any point on the edge some small value $\epsilon \approx 10^{-}13$ along the direction of steepest descent. If this point is outside the polyhedron, the corresponding edge is a downhill edge. Downhill edges are divided according to which neighbor they intersect. Then we compare the set of shared edges with neighbors to the set of 'downhill' edges. The original polyhedron is then split into multiple polyhedra upwards along the direction of steepest *ascent* according to the various neighbor edges. This process is repeated multiple times since partitioning of downhill polyhedra changes the neighbors and requires more partitioning to ensure no more than one downhill polytope.

5.2.3 Graph Traversal Algorithm for Watersheds

As each polyhedron leads to no more than one other, we now have our directed pseudoforest. Since each polyhedron represents a node on a graph with no more than one outgoing edge, we can follow the paths on this graph structure to find a watershed, with a few extra rules specific to the geometric conditions.

Ravine Condition

When two polyhedra point to one another, they have some line connecting them that must be 'downhill' of both. The points on these polyhedra then would normally flow down along this 'ravine' line until they reach another polyhedron. In the case where two polyhedra point to one another, we must go back to the geometry and find the new polyhedron they then point to. We call this a **ravine condition**. In order to determine the result of the ravine condition, we use the lowest point on the 'ravine' line r_{-}. If this point is contained in one other polyhedron, the two polyhedra then have their corresponding outward graph edges reassigned to this new polyhedron.

It is possible that this point will be contained in more than one polyhedron and one must be chosen to proceed. We choose the polyhedron with the steepest downhill slope, as it represents the most likely direction for the motion to proceed in should some instability be introduced (as it often is in real cases).

Base Watersheds

After applying ravine conditions, certain geometric cases present **base watersheds** by having no further polyhedra to which they will flow. There are two cases in which we can declare a base watershed reached. The first case is one in which the polyhedron points to nothing, or equivalently the vertex on the graph has no outgoing edge. Polyhedra which satisfy this condition are assigned to a single 'outside' watershed. The second occurs when polyhedra point to each other in a loop containing more than two polyhedra. In this case we have reached a set around a single watershed

point, and in this case these polyhedra are sides of a single minima.

Once the base watersheds and ravine conditions have been integrated into the graph, each remaining polyhedron is assigned to a watershed.

At this point all but the base watersheds are unassigned. For each unassigned polyhedron, we traverse down the graph along corresponding nodes until we reach a base watershed, or a polyhedron (node) that is already assigned. We assign this polyhedron and each polyhedron along its path to the corresponding watershed.

Once all polyhedra have been assigned, we can easily check to find the set of polyhedra in the same watershed as our target configuration. This set of polyhedra represents the 3D surface corresponding to the area of acceptance. To find the final AA in the correct dimensions, we project this down into the free DOF - in the 2D case, x and θ.

5.2.4 Results: Pre-Partitioning Method

As a tool to evaluate connector shapes, the method is broadly applicable to any connector shape. However, we chose two 2D connectors - the V-Face and X-Face (Fig. 3.1) which were analyzed using this method. We examine the two design parameters mentioned in Section 2.4 - Aspect Ratio (height of the connector geometry) and Center of Rotation (location of the point about which the connector rotates).

These results are listed as an area value in Tables 5.2 and 5.3, with the plots showing the final shapes in Figures 5.8. We can see that most of the areas of acceptance for the X-Face are larger than the V-Face by a factor of up to two, as expected. This shows preservation of relative areas of acceptance between the two connectors as found in Section 3.1. Contrary to expectations, four values in the upper right section of the table are larger for V-Face than X-Face.

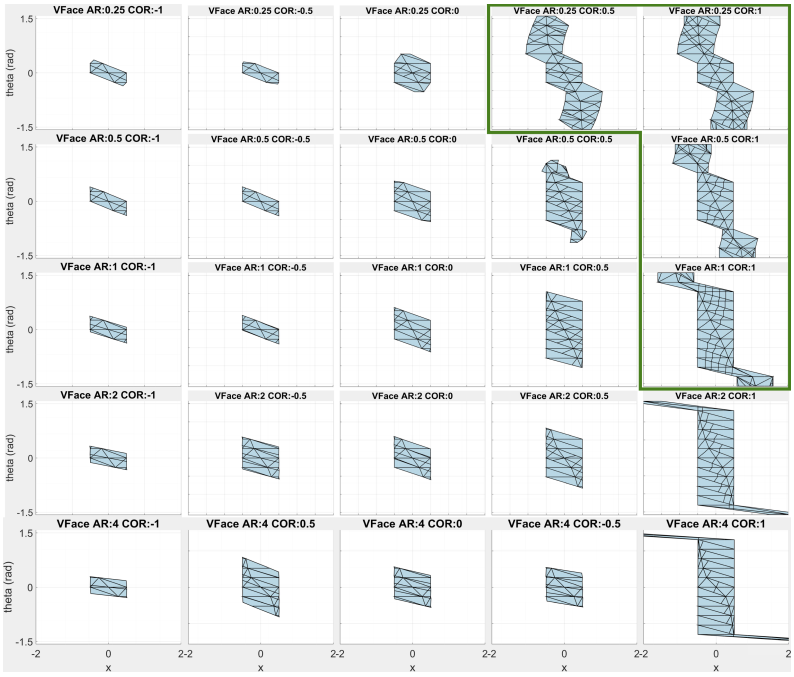
This may be due to the change from dynamic to quasi-static analysis. Several large patches reach critical points which in the dynamic case would pass into the area of acceptance, but in the quasi-static case do not. The results show that this

method is capable of determining the area of acceptance within a certain level of accuracy.

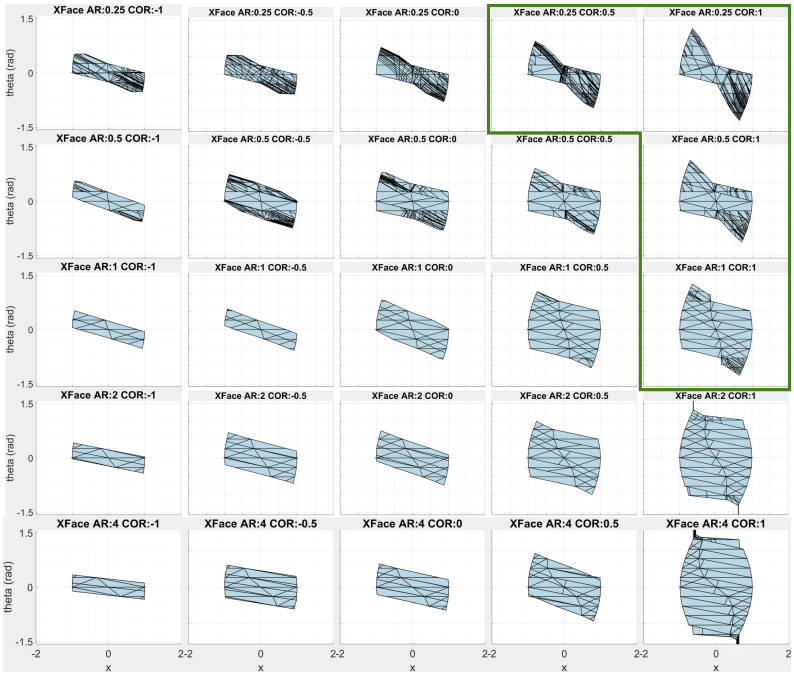
Several trends are observable in terms of the design parameters. More remote (positive) COR tends to correspond to larger AA, as does smaller aspect ratio. We desire lower aspect ratio in order to keep connector size small for a given module width, so we find this result encouraging.

Computational complexity is difficult to determine exactly, though it was observed that the most expensive part of the operation was building a graph detailing which polyhedra were neighbors. To determine if two polyhedra are neighbors, we simply perform an intersection operation and check to see if it is empty. However this operation must be performed once for each pair, resulting in $\mathcal{O}(P^2)$ intersection operations where P is the number of polyhedra on the surface. The number of polyhedra on the surface is proportional to the number of edges of the two connectors E and the number of slices N. We therefore estimate the computational complexity of the pre-partitioning method is at least $\mathcal{O}((EN)^2)$. Note that including more rotational degrees of freedom makes this worse, as the number of polyhedra becomes $EN^(r)$, where r is the number of rotational degrees of freedom. In the full SE(3) with 3 rotational degrees of freedom, the computational complexity would likely be at least $\mathcal{O}(E^2N^6)$.

Some limitations to the method exist. Complexity scales up very quickly with the number of rotational degrees and resolution, as well as the number of connector edges. Geometrically complex connectors with lots of edges will take longer to analyze. Higher rotational resolution and increasing the number of rotational dimensions also increases the time required, making the method somewhat impractical at higher dimensionality.



(a) Areas of acceptance for V-Face



(b) Areas of acceptance for X-Face

Figure 5.8: Results of the Pre-Partioning Method. All AAs are transformed such that the y-axis lines up with the center of the face at each angle

Table 5.2: Pre-Partitioning Method AA computed for V-Face

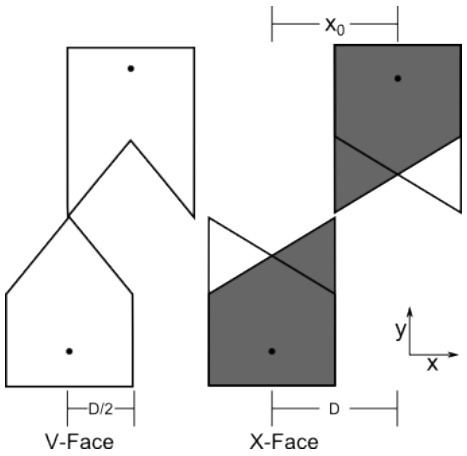
	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.38279	0.36586	0.85202	3.01069	3.14159
AR:1/2	0.40842	0.41129	0.86089	1.70012	2.77135
AR:1	0.43884	0.41643	0.87031	1.84022	3.14159
AR:2	0.45084	0.88724	0.84957	1.34941	3.14159
AR:4	0.46520	0.93465	0.88223	1.24182	3.14159

Table 5.3: Pre-Partitioning Method AA computed for X-Face

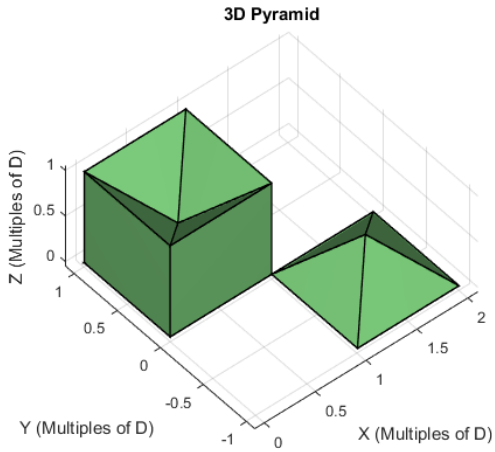
	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	1.09399	1.13635	1.28541	1.48352	1.84839
AR:1/2	0.97134	1.71965	1.99431	2.09141	2.27377
AR:1	0.92277	0.92203	1.79100	2.90494	3.13473
AR:2	0.88932	1.74696	1.66098	2.79853	4.09213
AR:4	0.89578	1.76955	1.65310	2.29481	4.65092

In practice, runtime for this method turned out to be prohibitively long. For the V-Face connector with a resolution of 12 slices on an average desktop machine (3.40GHz), a runtime of 48.6 minutes was observed. Higher resolutions were observed to take even longer, with resolutions of 21 taking nearly 8 hours to complete. Higher dimensional tests with the 3D connectors were attempted (i.e. checking errors in (x,y,$roll$) or (x,$roll$,$pitch$)) but these were found to be prohibitively long - calculations had to be terminated after a day and a half of runtime.

5.3 Discrete Representation Method



(a) Planar V-Face and X-Face at maximum x-offset. Note the X-Face is a connector composed of two separate mating layers.



(b) Pyramid connector pair geometries: concave on left, convex on right. Pyramid flipped upside-down for visibility.

Figure 5.9: Three tested connector geometry sets for the discrete method

The discrete method [24] for determining area of acceptance differs from the pre-partitioning method in several ways. First, rather than representing the contact space as a set of polytopes in space, we build a grid of points sampled directly from the configuration space slices. This saves us time calculating the higher-dimensional

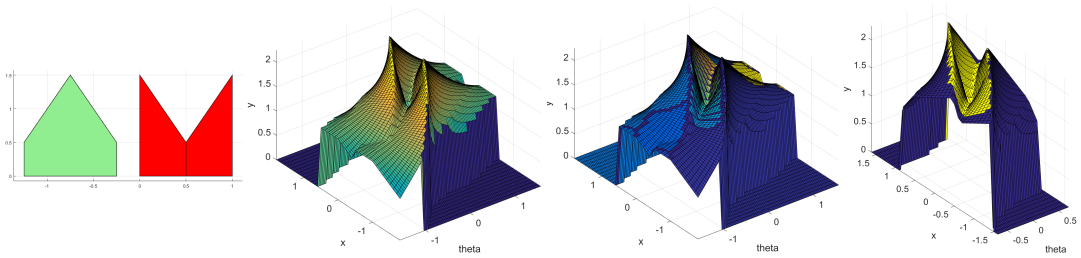


Figure 5.10: Full process for determining AA using discrete method. From left to right: V-face connector geometry, Configuration space height map (color corresponds to representative height), initial flooding results(color by initial watershed assignment), post-merge watershed/AA (final AA in yellow, outside area truncated)

obstacle, since we do not have to perform the CPC algorithm (Algorithm 1 or do the other operations on polytopes in that high-dimensional space. Second, the grid does not require partitioning or processing prior to a graph search for the watersheds. Third, we can employ proven watershed algorithms such as Meyer's flooding algorithm [45] to determine the watersheds and thus the area of acceptance. This method has five steps, which are each detailed and discussed below.

D.1 Define connector geometries

D.2 Generate Contact Space grid from Minkowski sum of slices

D.3 Initial watershed assignment by flooding

D.4 Merge watersheds by depth

D.5 Assign dam points to get final assignment

Defining connector geometries works the same as in 5.1.1. While the method in Section 5.2 allowed us to use exact (not sampled) representations of each slice, runtime was prohibitively long for higher dimensional cases or slice resolutions. In this method we instead use a discrete method common within image segmentation which runs faster at the cost of inexact representations of each slice. In the first

method, the partitioning required multiple calculations on large sets of convex polyhedra, significantly increasing computational cost. Even simple geometries like the V-Face and X-Face generated a very large number of polyhedra when using the CPC method for representing rotations. Computational run-time constraints limited practical time analyses to no more than three error dimensions with standard desktop computers. The flooding method used on the grid surface, however, is known to be $O(N^2)$, and in practice was observed to be many times faster than the partitioning method. Exact measures of time saved are detailed below in Section 5.3.2. This method uses a grid-based sampling of the configuration space obstacle rather than the complete polyhedral representation. This has two major advantages. First, we no longer have to approximate the space between slices or keep track of the full obstacle geometry. Instead, we can generate each slice, sample the requisite points, and then discard the exact geometry. Second, we determine the watersheds by use of well-understood flooding algorithms on these discrete grids developed for image segmentation. Collectively these advantages lead to an apparent savings in terms of computation time over the previous geometric methods.

For ease of use and adaptability, we again used the Multi-Parametric Toolbox [36] to generate the slice P-collections as well as perform the Minkowski operation necessary to generate the obstacles.

A configuration space obstacle (C-obstacle) for a robot capable of only translation is generated by the Minkowski difference ($\ominus$) of the robot and obstacle. In order to analyze the configurations including rotations as in the pre-partitioning method we rotate our robot A, successively through a range of possible rotations $[\theta_1, \theta_2, \ldots, \theta_N]$ and generate 'slices' $A \ominus B(\theta_i)$ for $i = (1, \ldots, N)$ at each discrete rotation.

Each θ_i slice is then sampled by taking it's 'top' value at a set number of points $[x_1, x_2, \ldots, x_N]$ in the translation dimensions (ie. x for 2D, x and/or y for 3D). These 'top' values are placed into a matrix grid of the appropriate dimension to form a height map over the potential offsets. If we consider the entire 'top' surface of a slice

for a given rotation to have the label S_θ, then in SE(2), the grid height values are as $h(x, \theta) = S_\theta(x)$ and in the constrained 3D set $\mathbb{R}^3 \times SO(3)$ as $h(x, y, \theta) = S_\theta(x, y)$. Example slices at $\theta = 0$ for the 3D connectors are shown in Fig. 5.4. For the sake of simplicity we assume the connectors are not capable of successful alignment beyond rotational errors of $[-\frac{\pi}{2}, \frac{\pi}{2}]$. The set of θ values for N slices is therefore $\theta_i = \frac{i}{N}\pi - \frac{\pi}{2}$ for $i = (1, \ldots, N)$.

Watershed Flooding Algorithm

Watershed methods on 2D discrete grids have been widely studied as part of the morphological segmentation problem in computer vision and other fields. Meyer's flooding algorithm [45] is the paradigm for initial segmentation - though other variations on the method exist [58] such as Priority-Flood [5] for improved speed.

We can use any flooding algorithm from the literature to determine the watersheds. Neighbors are considered 8-connected in the two-dimensional case and 26-connected in the three dimensional case. We chose in this work to focus on determining of the practicality and performance of this *type* of algorithm, so we use an implementation based on Meyer's flooding algorithm, due to its relative simplicity of implementation. Other algorithms appeared to be more difficult to implement and carried no other uniquely helpful attributes in this case.

Our implementation of flooding works as follows:

1. The grid is searched for local minima, points which are strictly lower than or equal to their neighbors.

2. Unique neighbors of each minima are placed into a priority queue based on height.

3. The lowest point in the queue is assigned a label based on its neighbors - if all have been assigned the same watershed value that point is assigned that value, otherwise it is labeled with the 'dam' value.

4. All non-marked neighbors are added to the queue.

5. Continue to assign values to the points as in 3 until all points have values assigned.

This algorithm completes quickly as each point only needs to be checked once, regardless of grid size or dimensions.

Merging Watersheds

Identification of watersheds is often insufficient. Due to the discretized nature, the representation of the configuration space obstacle geometry is not perfect. It is possible for paths to a minimum (e.g. docking condition) to be lost as a result of unfavorable sampling locations. For example, a narrow canal on the surface oriented diagonally with respect to the sampling dimensions can appear as multiple smaller watersheds when sampled in a grid (See Fig 5.11). Higher resolutions result in fewer sampling errors, but there is no point where errors are guaranteed to vanish for an unknown shape. A sufficiently narrow and steep 'ravine' will still generate small artifacts of the type seen in Fig 5.11.

This problem is similar in nature to the 'over-sampling' problem from the morphological segmentation literature. Multiple methods exist for automatically merging over-sampled regions, such as metric-based merging [8] [42] and hierarchical or multi-scale segmentation [6]. In metric-based merging, a metric value is assigned to each watershed (based on depth, volume, surface area, or some mix thereof). Watersheds below some threshold are then merged with their neighbors. Hierarchical methods construct a higher-level graph of the oversegmented watersheds and perform another flooding algorithm on this graph to determine the higher-level watersheds, resulting in regions that are nearly homogenous but still clearly separate.

We have chosen to use a metric-based merging technique, again for simplicity, wherein watershed regions of sufficiently low depth are merged with a neighboring

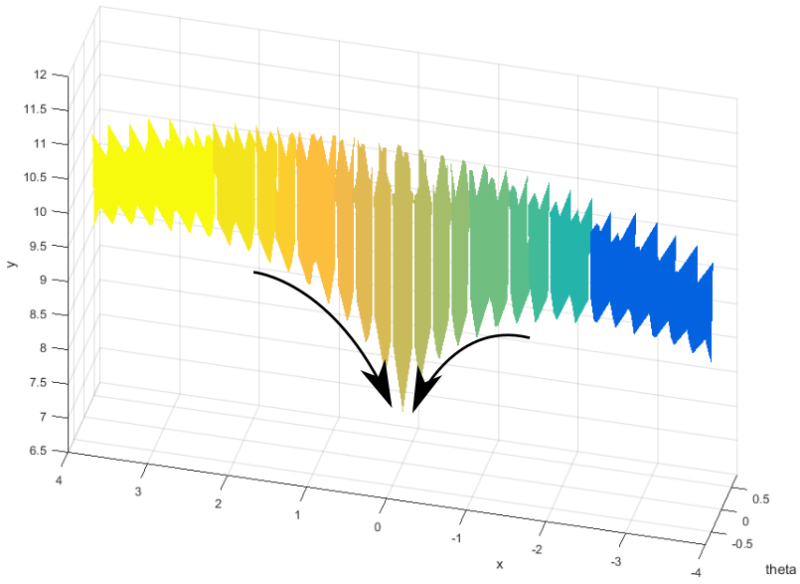


Figure 5.11: Example of sampling artifacts. Each color shade is labelled a separate 'watershed' after flooding. The true shape of the C-Space has a curve as indicated by the arrows.

region. Watershed depth is defined as the vertical distance between the lowest point in the watershed and the lowest boundary point. The neighboring watershed to be merged is chosen as the one that water would 'flow into' if the low depth watershed overflowed. This is the watershed neighboring the lowest boundary point. If a watershed has a depth value below the 'depth tolerance' it will be merged. In practice the depth tolerance value is found manually, by looking at the resultant depth values for each initial watershed and finding a clear dividing line. Acceptable depth tolerance values were observed to scale with height of the connector and the resolution. We found that a depth tolerance value of $\sim 0.1 \times AR$ worked well for most of our testing.

As a finishing step, points labeled 'dam' are assigned to the watershed of the neighboring point that is in the direction of steepest descent from the dam point. The total number of points assigned to the watershed of the target docking configuration, multiplied by the area of each point gives the final AA number.

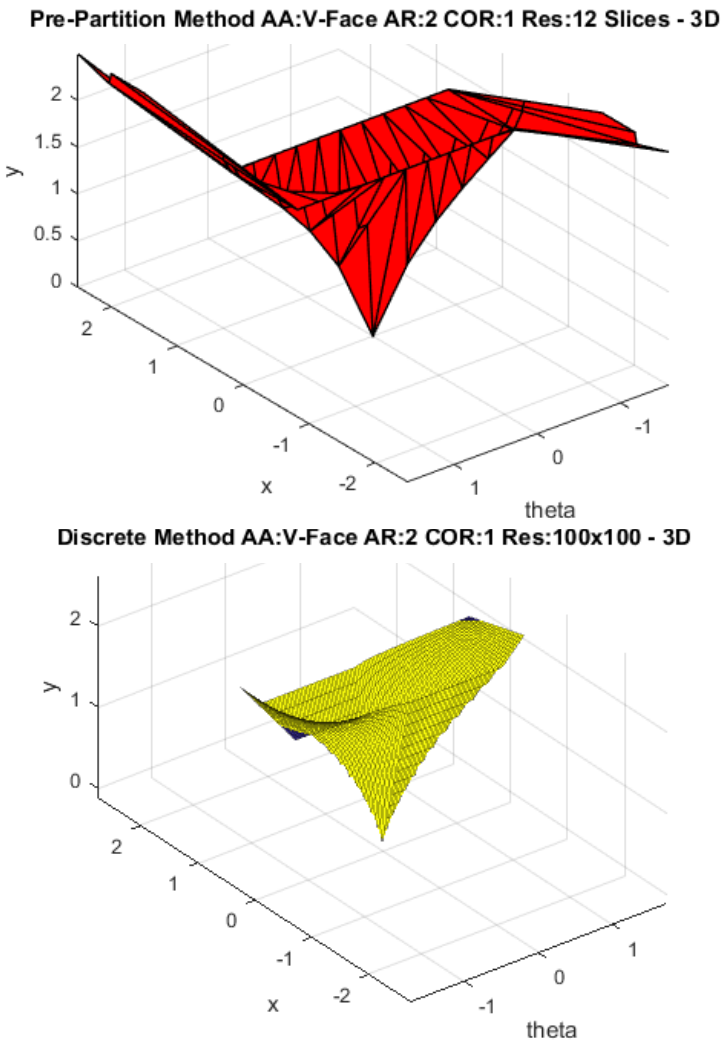


Figure 5.12: AA geometry for the same set of connectors found by partitioning (top) and discrete method (bottom). Higher resolution in the discrete method allows better determination of borders between watersheds, revealing a reduced AA compared to partitioning.

5.3.1 Determining ONAA

If you are sampling numerically, as we are in simulation and this discrete method, it is sufficient to assign each successfully docked point a number corresponding to the largest n-cube of successful points around it and take the maximum. An algorithm

for finding the largest n-cube within a sampled grid is included below. The only input that is required is a truth matrix T, on the same scale as the watershed data, where T = 0 at points not in the AA, and T=1 at points in the AA.

function Find ONAA ($\mathcal{T}$)

Input : N-dimensional truth matrix $\mathcal{T} \in \mathbb{I}^{d_1 X d_2 \ldots X d_n}$

Output: ONAA in $\mathbb{I}$

for $i_1 \leftarrow 2$ **to** d_1 **do**

  for $i_2 \leftarrow 2$ **to** d_2 **do**

    $\vdots$

    for $i_n \leftarrow 2$ **to** d_n **do**

      PrecedingIndices $\leftarrow$ All combinations of (i_j, i_{j-1}) for j=1 ... n

      $\mathcal{T}(i_1, i_2, \ldots, i_n)$=min($\mathcal{T}$(PrecedingIndices))

    end

  end

end

ONAA $\leftarrow$ max($\mathcal{T}$)

Algorithm 2: Algorithm for determining ONAA, runs in O(N) time, where N is the number of elements in the grid

5.3.2 Results

We used this method to analyze several sets of connectors - both in 2D and constrained subsets of SE(3). Matrix resolution for the 2D connectors is 100×100, and for the 3D connectors is $30 \times 30 \times 30$. AR and COR values tested are the same as for the pre-partitioning method;AR:[1/4 1/2 1 2 4], COR:[-1 -1/2 0 1/2 1] .

With the improved runtime of this new method we can now feasibly examine 3D connectors in higher dimensional error spaces, and 2D connectors at much higher resolutions. We found the AA of four different sets of connectors. Two of the sets (the V-Face and X-Face seen in Fig. 5.3) are defined in 2D and are allowed to move in SE(2). This represents a significant improvement in resolution over the previous

method while still saving computational time. Another set (the 3D Pyramid in Fig 5.4) is defined in 3D and analyzed in a constrained subset of SE(3). This constrained subset consists of the approach direction z and combinations of three of the possible offsets(x, y, *roll*, *pitch*, and *yaw*). Comparisons of typical results from the Pre-Partitioning Method vs. the Discrete Method can be seen in Fig. 5.12.

We tested and timed each method on a 3.30GHz processor for the same connector pairs and offsets in 2D. The partitioning method was observed to take 48.6 minutes with a resolution of 12 rotational slices for the V-Face connector. The discrete method was observed to take 3.18 minutes for a resolution of 100x100 points (translation and rotation, respectively). The discrete method was therefore 14 times faster and allowed an 8 times improvement in rotational resolution. As the effect of resolution on the results was not yet known, we initially selected a high resolution identical in each dimension for initial testing. A full analysis of the improvement in computational complexity is planned for future work. Points sampled are spaced evenly - in translation along the range of positions where the connectors might make contact (determined below), and in rotation between $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

Table 5.4: Sum AA computed for V-Face

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.444	0.557	0.971	3.332	3.33
AR:1/2	0.444	0.615	0.936	3.03	3.33
AR:1	0.622	0.738	0.966	2.17	3.06
AR:2	0.0339	0.881	0.896	1.57	3.23
AR:4	0.0465	0.0440	0.886	1.15	3.28

Table 5.5: Sum AA computed for X-Face

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.776	0.936	1.22	1.60	3.15
AR:1/2	0.768	0.899	1.40	1.97	2.58
AR:1	1.04	1.20	1.35	2.52	3.76
AR:2	0.730	1.35	1.76	2.51	4.77
AR:4	0.0767	1.07	1.95	2.31	5.17

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.21237	0.28274	0.40715	0.60821	0.7854
AR:1/2	0.15205	0.21237	0.3217	0.60821	0.84949
AR:1	0.10179	0.15205	0.21237	0.40715	0.84949
AR:2	0.061575	0.080425	0.12566	0.28274	0.84949
AR:4	0.031416	0.045239	0.061575	0.15205	0.84949

Table 5.6: ONAA computed for V-Face

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.66476	0.7854	0.66476	0.66476	0.66476
AR:1/2	0.50265	0.66476	0.91609	1.0568	1.0568
AR:1	0.28274	0.45365	0.66476	1.2076	2.0106
AR:2	0.15205	0.2463	0.40715	0.84949	2.5447
AR:4	0.061575	0.10179	0.18096	0.45365	2.5447

Table 5.7: ONAA computed for X-Face

2D Results

In Tables 5.4 & 5.5, values are in dimensionless units. Translation units are fractions of the connector width D and rotational units are in radians, so AA units are equivalent to (connector widths $\times$ radians). While exact methods for finding the AA to test our method against are currently unknown, we can compare to results from the pre-paritioning method to determine if they are somewhat in agreement. With few exceptions, the new results are within 80% of the results from partitioning, and typically larger. More than half are within 20%. The exceptions are results in blue in the high-AR\negative-COR range which are $< 10\%$ of partitioning results. This combination of parameters is particularly bad for the 2D connectors. The AA is so small that it is difficult to accurately measure at the resolution used for the

partitioning method (N=12). Since it was not able to represent these AAs in sections less than the N=12 resolution, the result ends up larger than is accurate.

The discrete method is more accurate than the partitioning method for two reasons. First, we make no assumptions about the nature of the surface between slices. Second, errors are a function of resolution and computational savings allow for us to run the method at higher resolutions.

If we look at the ONAAs we can notice a few points of interest. First, the AR and COR trends are mostly preserved compared to the Sum AAs. The largest values are still at the highest AR and COR, though most of the COR columns are best suited with low AR. Second, the ONAA is (as expected) smaller than the Sum AA, to a large degree. In each case, the ONAA is limited by the maximum translation acceptance, as a result reaching a maximum in the right hand column.

3D Results

Results for 3D sets (*x*,*y*,*pitch*), (*x*,*pitch*,*roll*), (*x*,*pitch*,*yaw*), and (*x*,*roll*,*yaw*) can be seen in Tables 5.8-5.11 As in the 2D case, the following values are in dimensionless units. Translation units are still fractions of the connector width D and rotational units are in radians. So for example in the (*x*, *y*, *pitch*) case, units are (connector widths × connector widths × radians). The largest AA in each table is highlighted in green, while the lowest is highlighted in red.

Tables for (*x*, *y*, *yaw*) results are not necessary as they are not affected by the design parameters. AR acts only on relative heights and so only scales the C-Space obstacle without changing the shape or watersheds. COR is measured along the *yaw* axis of rotation and so it also has no effect. The AA for the 3D Pyramid is 2.47. Some combinations of offsets have been omitted due to symmetry - for the two connectors chosen (*x*, *y*, *pitch*) is symmetric with (*x*, *y*, *roll*), and (*x*, *pitch*, *yaw*) is symmetric with (*y*, *roll*, *yaw*) for example.

The tables show that a more positive COR generally means a higher AA. This is

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.123	0.248	0.55	1.23	1.56
AR:1/2	0.187	0.424	0.663	1.54	1.77
AR:1	0.211	0.531	0.761	1.51	1.98
AR:2	0.476	0.626	1.23	1.8	2.08
AR:4	0.579	0.795	1.23	1.99	2.01

Table 5.8: AA for (x,y,$pitch$)

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	0.134	0.785	3.14	3.31	1.14
AR:1/2	0.486	0.873	2.52	4.06	3.80
AR:1	0.126	0.333	0.908	3.91	5.19
AR:2	0.0331	0.0888	0.266	1.49	6.05
AR:4	0.00398	0.00398	0.0199	0.325	6.65

Table 5.9: AA for (x,$pitch$,$roll$)

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	2.12	2.65	3.62	3.91	3.71
AR:1/2	1.06	1.65	2.34	3.91	4.03
AR:1	0.576	0.649	1.25	3.52	3.74
AR:2	0.13	0.335	0.603	1.17	3.74
AR:4	0.0994	0.111	0.121	0.649	4.31

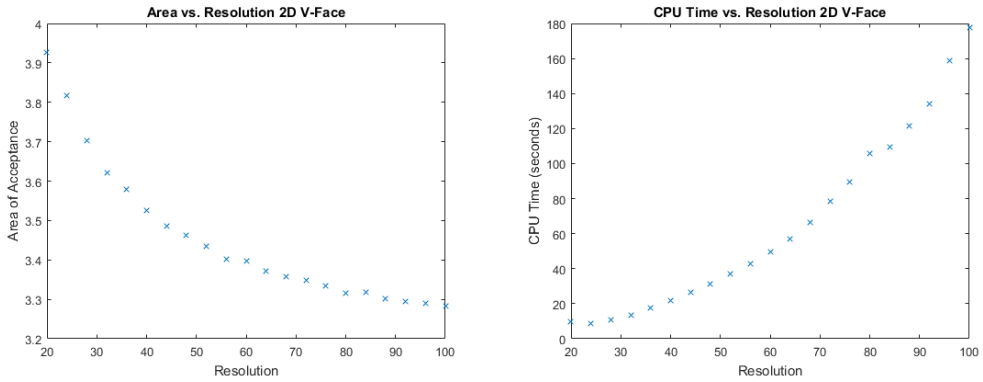
Table 5.10: AA for (x,$pitch$,yaw)

	COR:-1	COR:-1/2	COR:0	COR:1/2	COR:1
AR:1/4	2.1	2.45	2.71	2.95	2.59
AR:1/2	0.892	1.44	1.68	3.74	2.96
AR:1	0.563	0.873	1.33	2.19	2.88
AR:2	0.368	0.421	0.706	1.83	3.1
AR:4	0.0437	0.0557	0.209	0.904	3.69

Table 5.11: AA for (x,$roll$,yaw)

not surprising since it is a more remote center of rotation, a known desirable property for docking systems. Higher AR often increases AA for high COR but decreases AA for negative COR. Lower AR values yield AA that vary less with COR. The (x, y, $pitch$) case is different from the others in that the minimum occurs at low AR rather than the high AR. Higher AR corresponds to faces that point more in the x and y directions, so they are likely better suited to cases focused on translation. In Table III, (x, y, $pitch$) the highest AA value is not the highest COR and AR (in the bottom right corner). This appears to be within the discretization error. If a point is sampled on one side or another of the border it can affect the AA slightly at that point. The difference between the bottom-right value and the largest value is equivalent to 42 points, which is within our margin of error. These areas are a few thousand points in size, with borders on the order of several hundred. When the geometry changes with the design parameters, sampling may cause a shift in how the 'dam' values are assigned. These can produce a sort of cumulative sampling error in the final values.

5.3.3 Convergence and Computation Cost



(a) AA vs. Resolution convergence. (b) CPU Time vs. Resolution

Figure 5.13: Convergence tests, 2D V Face AR 1/4 COR -1/2

Test runs were performed at multiple resolutions to determine the effect on AA and computation time. We can see from Figure 5.13 that for the 2D connectors, as the resolution N increases, the AA slowly converges and CPU time goes up approximately as a factor of N^2. The CPU time result is as expected - the watershed algorithm is well known to be on the order of the number of points, or in this case, $\mathcal{O}(N^2)$. Note that the AA starts larger and slowly decreases due to a characteristic of the method - points normally used as 'dams' are assigned based on steepest descent - often being assigned to the AA, resulting in slightly higher values. Since the size of these cells is larger at lower resolutions, the AA is slightly higher. As the resolution increases, the estimation nears the true value.

5.3.4 Experimental Testing and Validation

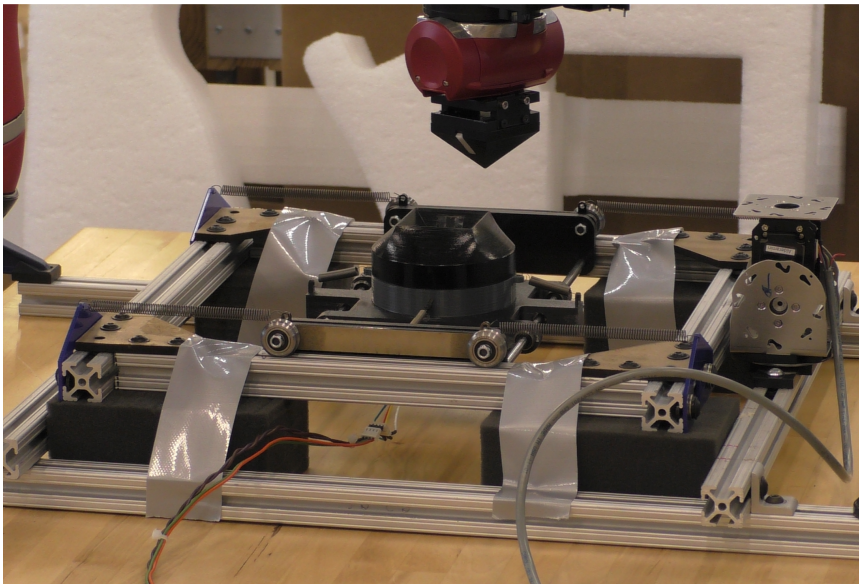


Figure 5.14: Connector testing stage and Sawyer robot arm with attached connector. Base connector window contains sensor

Testing of the (x, y, yaw) results for the 3D Pyramid was performed on a Sawyer 7-DOF robot arm. The testing setup can be seen in Fig. 5.14. The Sawyer arm is accurate to within +/- 0.1 mm [1], and the connectors were built at a scale of 2"x2"x2". The top connector has a white stripe on two of the four pyramid faces to help sense successful alignment. The bottom (base) connector was on a sliding stage capable of separate movements in x, y, and yaw. Each degree of freedom has a spring return to ensure a successful return to the zero position. Additionally, a single CKBot servo module [18] was placed at the edge of the stage to perform perturbations between tests. The perturbations allowed the sliding stage to overcome static friction, easing return to the zero position when lubrication alone proved insufficient.

An infrared (IR) emitter-detector pair is placed inside the base connector, facing a clear plastic window to the alignment site of the connector attached to Sawyer robot. This sensor was in turn connected to an Arduino Micro board, which read the

value of the sensor and recorded the value to a connected laptop PC which was also issuing instructions to the Sawyer robot and CKBot module. Testing determined very different IR values resulted depending on the final state: high values if the white section of the connector faces the sensor, low for black, and low-middle if the connector is not in the aligned position. During testing the black faces never reached the sensor but as this is a possible final condition (results from more than 45 degrees *yaw* misalignment), we must be able to detect it to distinguish it from the other states.

The arm was programmed to attempt alignment at offset points inside the AA to just outside the boundary, such that the tests would terminate at the first expected failed alignment. These tests were restricted to *yaw* offsets of 30 degrees or less. With *yaw* conditions beyond that friction and material deformation caused the connectors to jam rather than successfully align.

We used this system successfully to test 150 points (cube with 5 points in x, 5 in y, and 6 in *yaw*) on the interior of the AA up to the boundary for the (x, y, *yaw*) set. Points were tested only in the positive octant due to natural symmetry of the AA.

These results show an agreement between the expected and actual results on the robot platform. The boundary appears at 1 inch from the center (as expected given the connector size of 2"x2"x2"). Results not taken beyond 30 degrees of *yaw* due to the jamming of the connectors. These results confirm the results of our method for the (x,y,*yaw*) case within the bounds we are able to test.

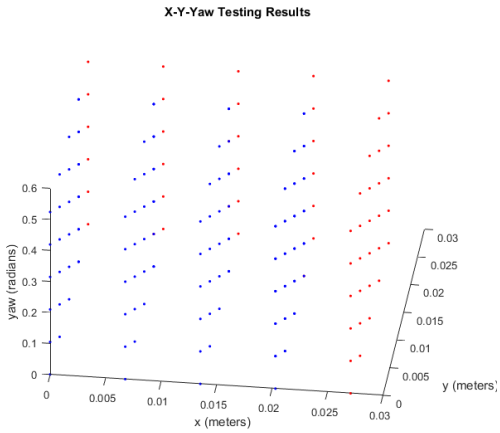


Figure 5.15: Results from stage testing plotted in 3D

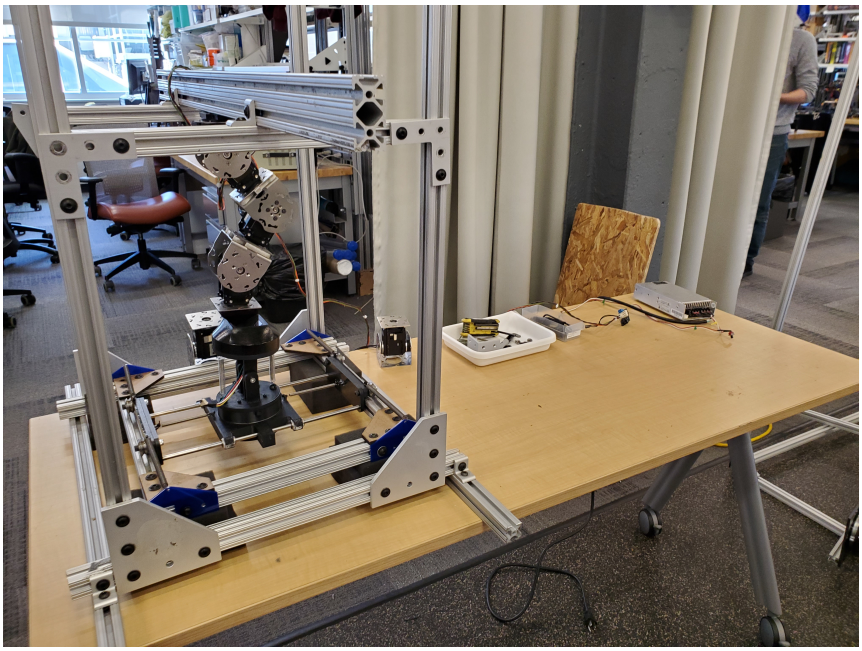
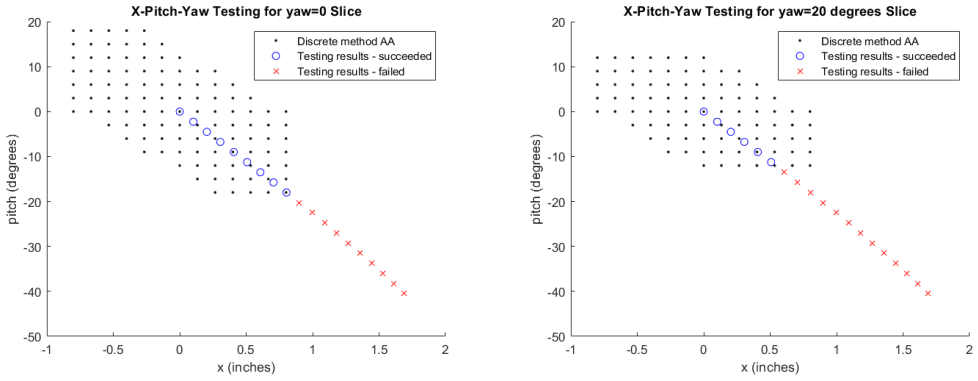


Figure 5.16: Connector testing stage and CKBot arm with attached connector.

Testing of the (x, *pitch*, *yaw*) results for the 3D Pyramid was performed on simple arm made from CKBot [18] robots, using a similar sliding stage to the previous test. The testing setup can be seen in Fig. 5.16. The connectors are identical to the previous test. The arm was calculated to be accurate to within $\pm$ 1.2 mm. A simple

script was writen to move the arm up and down along the rotation point of the end module. As a result, the angle of the end module corresponds to a change in both x and *pitch*. Testing was performed at 20 different angles from 0 to 45 degrees, with 2.25 degrees between each step.

This process was performed at *yaw* angles of 0 and 20 degrees, respectively. Beyond 20 degrees, the connectors have a tendency to jam; though the simulation data shows that they would still succeed almost as readily if the friction were less significant. The plots in Figure 5.17 show the accepted points for each '*yaw* slice' in both simulation and experiment. Points which failed in the experiment are provided as well, for reference. These plots show that the alignment process fails where predicted to within the resolution of points tested. As a result, we can say that the discrete method appears to be verified by these experimental results.



(a) First test

(b) Second test

Figure 5.17: Testing results with simulation for comparison

Chapter 6

Design and Control of Attachment Mechanisms for High-Alignment Connectors

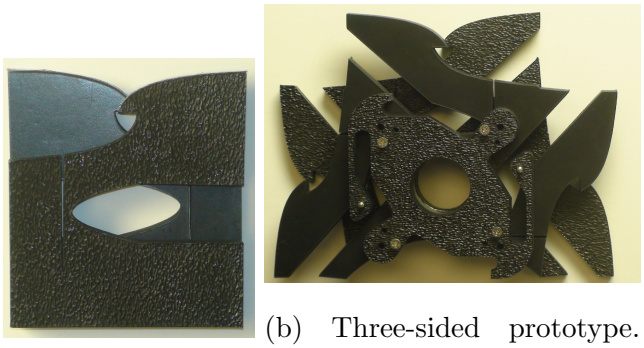
In order to have a useful docking connector, we must not only look at alignment, but also the process of attachment and detachment. As we saw in Chapter 1 here are many different ways of attaching two modules or rigid components including fasteners, latches, magnetic forces, and electrostatic forces [32] [29] [61]. Electrostatic and magnetic connections often have a relatively low ultimate strength, which can limit their capabilities to form large or extended conglomerates. Fasteners are very strong but often difficult to use autonomously. Autonomous attachment by fasteners has been accomplished [44], however presently each fastener requires its own framing and actuator. Latching or mechanical capture systems such as DARPA TEMP (See Figures 2.6 and 2.4) or M-TRAN [34] tend to be strong, and reliable in use once alignment is established. We believe we can combine strengths and achieve robustness by composing alignment with latching.

We extend the idea of the X-Face connector to include a passive latching mechanism for reconfiguration. It is in general desirable for a latching system to require

not only a small force to latch, but a large force to induce failure - the more modules a latch can support, the more configurations are available to the system. We start with a two-dimensional connector, the 2D X-Face, as a proof of concept.

In order to obtain a relatively high ultimate strength while making attachment and detachment easy, we performed an examination of latching mechanisms. These mechanisms were examined in combination with high-alignment geometry in order to maintain the advantages of those mechanisms. Several prototypes were constructed and tested to collect data about the relevant properties of these latching mechanisms.

6.1 The 2D Latching X-Face



(a) Single sided prototype

(b) Three-sided prototype. ModLock [16] (shown on top) connects to CKBots.

Figure 6.1: Latching X-Face prototypes, close view.

A mechanism for use on in-plane (2D) modular robotics applications was presented at IROS 2014 [19], and can be seen in Fig. 6.1. We present here both a one-sided version of the connector in order to better understand the working principles, as well as a three-sided version for practical use with a CKBot chain.

The design goals for this new connector include: a latching functionality, low required force to actuate, and high strength when latched. Therefore, the connector has no actuated components per se; energy for the latching operation must be added

externally; often by the motion of the modules that hold it. Likewise, we keep the force required for latching low while retaining high bonding strength. To that end we will use as a performance metric, the **bonding ratio**, which is the force required to disconnect (in this case, material failure) divided by the force required to connect. This metric permits the evaluation of performance of connectors with different designs or parameters.

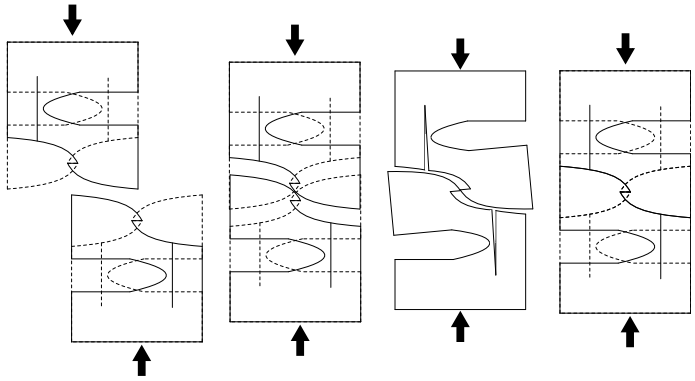


Figure 6.2: Connection sequence. Top layer-solid, bottom layer-dashed. Bottom layer is removed for clarity in the 3rd image.

As in the 2D X-Face, two layers are attached to a face of a module and then moved by the modules to be adjacent to a target module, forming a connection. One layer is a mirror image of the other. Faces are curved in order to change the direction of the contact forces slightly, allowing the compliant latches to open. The docking process for this design can be seen in Fig. 6.2 and follows:

1. Modules move in the plane to the target locations.
2. Connectors align by combination of module actuation forces and the mechanical connector interaction forces.
3. Modules are pressed closer together, forcing the compliant features outward.
4. Compliant features snap into the latched position to hold the docking connection together.

When the connector is used on a chain of cube or square-shaped modules such as CKBot [16] that is moving in a plane, we might wish to have connectivity to neighboring modules in all four adjacent locations (front, back, left, and right). However, in an articulated chain, at least one degree of freedom is required between two faces. Rigidly connecting all four sides would prevent this. Therefore, for our planar connector prototype we use three latching faces and a profile swept out to allow relative motion.

One layer of one side of the design is shown, with important components labeled, in Fig. 6.3. These layers are rigidly attached to each other and the module *below* (in the y direction) the compliant features so that they still function as intended. This combination of features results in an ungendered connector which creates a strong mechanically latched connection without requiring actuators, but still has a high area of acceptance.

Prototypes can be seen in Fig. 6.1. The prototypes have a small separator layer between the X-Face layers to prevent interactions between those layers due to small deformations or misalignments. If the separator layer is equal to or greater than the width of the X-Face layers, the connectors can easily be disconnected manually by displacing them in the out-of-plane direction. The separator layer also prevents incidental interaction between layers due to small manufacturing defects or bending. We used 1/4" ABS lasercut plastic for all three layers on the prototypes. Prototypes for bonding ratio testing have one latching face while those used for reconfiguration demonstrations have three latching faces. These will be referred to as the 'one-sided' and 'three-sided' prototypes, respectively. We built the prototypes at a scale on par with CKBot, our target platform [16]. CKBot modules measure 65.5mm×65.5mm×90.4mm. The distance between rotors in a chain is 95mm, setting the center-to-center distance between the connectors. It is unclear what effect the additional compliance has on the area of acceptance, since that concept does not incorporate forces or compliance. Empirically however, we saw no significant changes

in the AA - when pushed together by hand the connectors docked as expected, even with significant errors in position and rotation.

Therefore on the three-sided prototype (Fig. 6.1b), the distance from the center of the overhang to the center of the shape which is placed over the rotor is 47.5mm, with the width of a latching face for the connector, 65.5mm.

6.1.1 Design Principles and Parameters

There are several qualitative design principles and corresponding quantitative design parameters which are essential to the function of this latching mechanism (refer to the labeled components of Fig. 6.3).

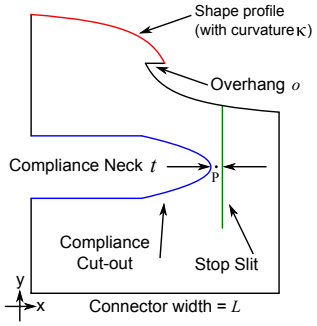


Figure 6.3: Latching X-Face with important design features.

Profile Maximum. Like the 2D X-Face, the highest point on the shape profile is at the outer edge. If we let the full width of the connector be L, and the width of the overhang be o, we can view the shape profile as a function $(f(x)|x \in [0, \frac{L}{2} + \frac{o}{2}])$. Then the function is at its maximum at $f(0)$. Furthermore, the profile function should be decreasing at all points: $f'(x) < 0 \forall x \in [0, \frac{L}{2} + \frac{o}{2}]$ to prevent jamming. The two layers permit full alignment in either direction. These characteristics ensure high area of acceptance, similar to the 2D X-Face. Note that the shape profile is rotationally symmetric about the overhang, so the male and female features of mating faces nest properly.

Overhang. Represented by the value o, the overhang prevents the connector from

becoming disconnected up to material failure; with the previous X-Face design, detachment was not prevented by separating motions along the mating axis. The width of the overhang determines how much deformation of the connector is necessary to perform latching, and thus affects the maximum force required.

Compliance. The shape and location of the compliant cut-out as well as the distance from the stop slit (which we designate t) determines the compliance. This distance t, is the thickness at the narrowest point of the section, which we call the *compliance neck*.

Curvature. We found that if the curvature κ is concave (which we will call positive curvature) or straight, the connectors had a tendency to jam.This was due to an interaction with the bent position of the latch arm with the shape. Once the arm begins to bend, forces change their direction and begin pointing away from the inwards alignment direction provided the curvature is non-negative. If the curvature is negative the rotation due to bending has a reduced effect on the direction of the interaction forces, ensuring a good alignment.

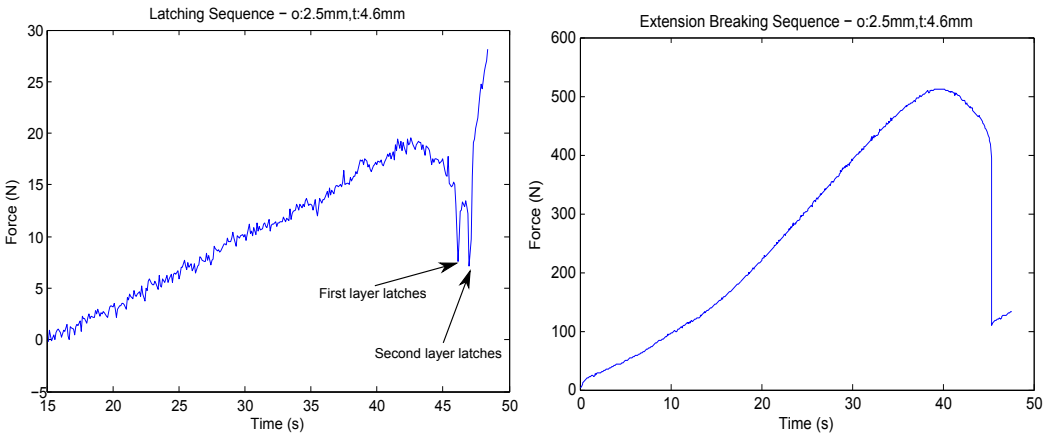
Mechanical Stop. Not only does the stop slit allow rotation of the necessary features, it prevents undesired rotation in the opposite direction. Without the mechanical stop section of the connector, the compliance would work in both directions, preventing rigidity under tension. The relevant quantities for the stop slit are the depth (y distance down from the curved face) and width (gap size in x). Width should be small. A wider stop slit allows too much deflection in tension. The stop slit should be fairly deep, ending below the geometry of the compliance cut-out as the thin section at the neck is essential for deflection. The location of the stop slit should be located sufficiently far in x distance from the rightmost end to ensure sufficient rigidity under tension. A distance of 15mm was found to be sufficient. A depth of 20.3mm was found to be sufficient for the compliance to work correctly. The gap width for the stop slit is equal to two times the laser cut kerf, or 0.36mm.

We model the compliant section by a pseudo-rigid body model (PRBM) [30].

This method equates the compliant mechanism to two rigid bodies connected by a torsional spring with an equivalent spring stiffness, and for simple cantilevers predicts the deflection path accurately within 0.5 percent [31]. The torsional stiffness for a simple PRBM model of this type is $k = \frac{EI}{l_{eq}}$ where E is the Young's Modulus of the material (our prototypes use ABS, E=2.1GPa), I is the area moment of inertia of the cross-section (assumed to be equivalent to a thin rectangular beam), and l_{eq} is the equivalent length. This model can be used to scale the system or adapt to changes in material properties and thickness. The equivalent stiffness for our three-sided prototype is approx. 4.99 Nm/rad . Since the corresponding relationship between deflection and moment is $M = k\theta$, overhang increases result in an increase in the maximum moment (thus force) required. The compliance neck thickness t affects directly the area moment of inertia I, as does the material thickness.

6.2 Force Testing

The parameters above are subject to design variation; as such it would be best to know how they influence the properties of the connector. To determine this relationship, we tested multiple versions of the prototypes with varying design parameters. In order to accurately measure the bonding forces of the connector, the prototypes were tested using an MTS Criterion™ Model 43 Testing System. Single-sided prototypes (including both layers) were mounted to a central piece which was then clamped into place on machine base and crosshead, respectively. Two types of experiments were performed; compression to determine the force required to perform latching, and latched extension until material failure to determine the load strength. Two typical tests are shown in Fig. 6.4. The latching for each of the two layers is visible in Fig. 6.4a. When a layer latches, a sudden dip in force can be seen. In this test the two layers latch one at a time allowing us to see both latchings at approximately 46 and 48 seconds. Force then quickly increases again as the assembly is



(a) Compression test showing typical test and noise characteristics. Successful latching of each of the two layers is visible.

(b) Tension break test: Deformation is visible before failure.

Figure 6.4: Force test plots

compressed.

For the prototypes constructed and tested, the profile maximum was kept at $x = 0$ primarily due to the 2D X-Face, which relies mostly on this feature for high area of acceptance. Prototypes were observed to have no difficulty correcting errors over approximately the same range of linear errors as the 2D X-Face when the one-sided prototype was connected by hand. This design parameter was not viable for testing in the MTS machine due to the lack of compliance.

The final design curvature was defined by an interpolated spline. The outermost end of the spline (at $x = 0$) was set to an angle of 35^0 with the x-axis seen in Fig. 6.3. Smaller angles tended to deflect without alignment; this counterproductively changes the direction of the force vector. It was found in testing of various angles that larger angles require more force to successfully latch.

To increase the torque applied throughout the curvature, the spline at the end touching the overhang was defined so as to 'point' directly to the rotation point

of the compliant section. In other words, if the spline was extended in a straight line from the overhang at the same angle it would intersect the midpoint of the compliance neck, labeled P in Fig. 6.3. This point is assumed to be the center of rotation for the flexure above based on observation of the flexing behavior. Other forces are approximately tangent to that line.

Since they both effect the force required for latching and thus the bonding ratio, six different connector designs with varying o and t, were tested. Results including compressive/extensive forces and spring constants as well as the bonding ratio can be seen in Table 6.1. We expect increases in o and t measurements to increase the forces required to connect. Increasing o requires increased deflection. Increasing compliance neck thickness t gives us a stiffer member. Both of these would result in larger forces. Prototypes with an overhang o of 1.27mm were constructed, but found to be too difficult to get a good latch - often the overhang would overlap less than 1mm, resulting in unstable connections. These prototypes were therefore excluded from testing.

The mechanical stop features were found to be sufficient to allow for compliance and rigidity, and not examined here.

TABLE OF STRESS TESTING RESULTS

o (mm)	t (mm)	Compression Force - Latching (N)	Extension Force - Failure (N)	k_C(N/mm)	k_E(N/mm)	Bonding ratio
2.5	2.3	6.18	489.11	9.674	84.76	79.1
2.5	3.0	8.56	666.39	44.19	92.63	77.8
2.5	3.8	10.42	669.48	27.58	125.7	64.2
2.5	4.6	19.57	512.32	69.37	141.8	26.2
3.8	3.8	14.55	824.03	25.20	147.8	56.6
5.1	2.3	9.32	508.77	7.526	86.02	54.6
5.1	3.0	16.36	646.74	26.01	108.0	39.5
5.1	3.8	22.34	813.09	30.78	149.2	36.4
5.1	4.6	46.58	960.10	91.22	134.6	20.6

Table 6.1: Varied Parameters Force Testing Results. Forces shown are the highest recorded during the relevant activity, which may be $\pm$ 1N. k_C and k_E are the spring constants in compression and extension respectively, starting from zero deflection.

Results in the table are somewhat unintuitive, but testing observations reveal more information. Designs with the smaller overhang ($o <= 2.5$mm) and a large neck ($t = 3.0$ or $t = 4.6$mm) were observed to have a different failure mechanism from the other connector prototypes. Most of the prototypes failed at the narrow band between the compliance cut-out and the stop slit (the compliance neck), breaking into two large pieces. The prototypes with $o = 2.5$mm and $t = 3.0$mm or greater failed at the tip of the overhang, deforming the material until the parts slipped apart. This change in failure characteristic is responsible for the seeming discrepancies in the Extension Force numbers. Note that the extension spring constant k_E increases even when the maximum force decreases for the fourth prototype. The change in failure characteristic causes failure at less force and distance, effectively weakening the material without effecting the spring constants.

Due to the change of the mode of failure, the effect of the compliance neck thickness t changed. With failure at the neck, a thicker neck predictably improved the strength of the connector. However, with failure at the overhang, a *thinner* (i.e. *more* compliant) neck improved the strength.

Greater compliance leads to greater deflection for a given force as expected. Deflection of the connector (now *clockwise* about P) helps prevent deformation at the overhang tip by changing the angle of the overhang relative to the force, allowing a stronger portion of the layer to take on the extension forces. The deflection also pushes the tips of the overhangs into the corresponding trench on the opposing connector, effectively acting like reciprocal hooks. When the overhang is held more rigidly (less compliance), the tips deform more easily as they share a greater portion of the force.

Otherwise, the data fits expectations - greater thicknesses increase both the force required to latch and the force required to break by extension. Effective spring constants increased with increased thickness as well. Interestingly, the bonding ratio decreased as the material got thicker - the force to connect increased faster than the

force to break.

For the three-sided prototype some alterations to the compliance cut-out were necessary to fit the 90^o profile as seen in Fig. 6.1b. These include narrowing the width of the compliance cut-out and changing the angle to fit the shape. As a result, the shape is slightly more compliant than would be expected of the same o and t measurements tested. So despite using the parameters $o = 5.1$mm and $t = 3.0$mm, this connector requires 9.61 N to latch and 584 N to disconnect, giving a bonding ratio of 60.9:1.

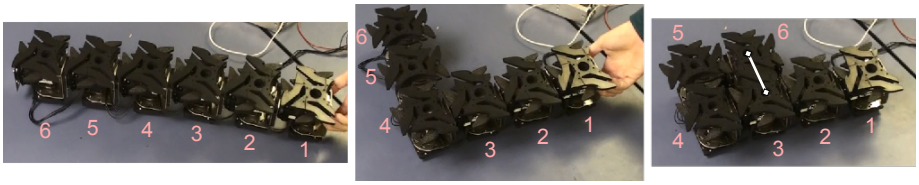
6.3 Reconfiguration testing

Verification of assembly as part of reconfiguration started from a preassembled planar chain of CKBot [16] modules. The three-sided version of the Latching X-Face Connector attached to one face of each module, with casters below to reduce friction.

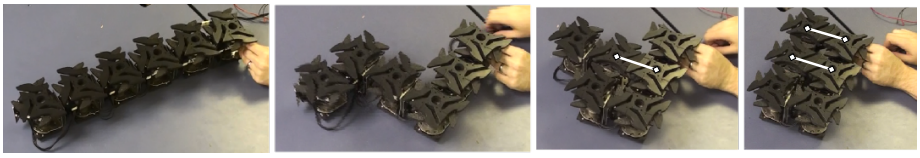
Reconfiguration was performed in a one-way fashion, that is the chain was able to attach faces but not detach. Detachment between experiments were performed by manually unlatching the connectors.

Three different reconfiguration sequences were tested and are shown in Fig. 6.5. The first example simulates a single lattice-type operation, which we call the 'P' shape. The second, called the 'Block' (essentially a 3×2 cluster of modules) starts with a straight 1×6 preassembled chain using ModLock connectors. It demonstrates a continuation of the 'P' shape lattice style reconfiguration; showing progressive latching (one module after another). This also demonstrates the capability to perform latching on mid-chain modules. Inductively, we assume more modules are capable of as much latching as required, done sequentially in a zipper-like fashion.

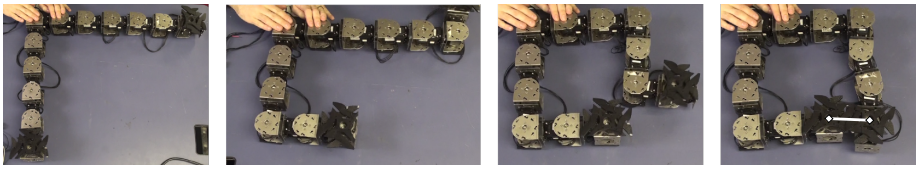
The third shape, a large square demonstrates chain style reconfiguration. This 12-module chain was assembled similarly to the others, but with connectors only at the chosen docking site. This is an important configuration because long chains



(a) P-shape (basic) reconfiguration



(b) Block-shape (lattice style) reconfiguration



(c) O-shape (chain style) reconfiguration

Figure 6.5: Reconfiguration sequences detail. Diamond-end white lines indicate latch connection sites.

of modules accumulate positional errors requiring large area of acceptance. When performing the attachment, we select kinematic arrangement that allows a large application of torque at the connection site.

The attachment process itself is a series of open-loop position commands. This command sequence requires some tuning to ensure a good latching connection, but once tuned correctly, completed 6 trials for each reconfiguration with 100% reliability.

The CKBot system uses EX-106+ Dynamixel Servos. with a stall torque of 10.9 Nm. The reconfiguration command sequences were optimized for speed; as such the actual application of force is not defined. However, the data has shown that any system which can provide a sustained force over 9.61 N should be able to successfully latch.

6.3.1 Latching Sequence

The latching sequence includes a set of positional commands to the servos as shown in Tab. 6.2. The servos use position control so while performing reconfiguration testing, commands go slightly beyond the nominal position of latching in order to apply appropriate force to engage the compliant members. For the two 6-module reconfiguration sequences, we refer to the modules in positional order starting from the closest to the control computer (i.e. module 1 is closest, module 6 is furthest).

In Tab. 6.2, for the single attachment in the 'P' shape reconfiguration, the first step of module 4 (at second 1) converts the chain to an L-shape that is slightly less than $\frac{\pi}{2}$. This means the connector will be slightly off-center in the next step at second 3 - the connector then only requires force to overcome the latch on a single layer, rather than both if it were centered. Once the first layer is latched by driving module 5 well past $-\frac{\pi}{2}$ at second 3, both modules are commanded to 'go slack', that is, apply no force. This allows the connectors to realign from the structural forces in the connector flexure. Before this slack motion, the two connectors are not aligned. Module 5 is then commanded past $-\frac{\pi}{2}$ once more to ensure the second layer latches. The configuration then goes slack once more to realign the faces.

A plot of the motions as sensed by the encoders in modules 4 and 5 is shown in Fig. 6.6. At second 3, the module moves past the nominal $-\frac{\pi}{2}$ corresponding to the latch of a single layer. At second 4, both modules go slack, allowing the connector forces to take over and realign both modules closer to the $-\frac{\pi}{2}$ position. Finally at second 5, module 5 uses the second motion command that completes the final latch. Second 6 makes both modules slack, to draw no power. The same sequence was repeated for the relevant modules in the 'Block' shape reconfiguration, and a similar procedure for the chain reconfiguration. This latching sequence demonstrates that these connectors are feasible for use with a modular system.

Here we have presented a new design for 2D latching in modular robots including high self-alignment and reliable latching. Design principles used to create the

CONTROL SEQUENCE TABLE

Time(s)	M4 Pos. Comm. (rad)	M5 Pos. Comm. (rad)
0	0	0
1	-1.67	0
3	-1.67	-1.74
4	GS	GS
5	GS	-1.74
6	GS	GS

Table 6.2: Control sequence for latching. GS stands for the 'go slack' command.

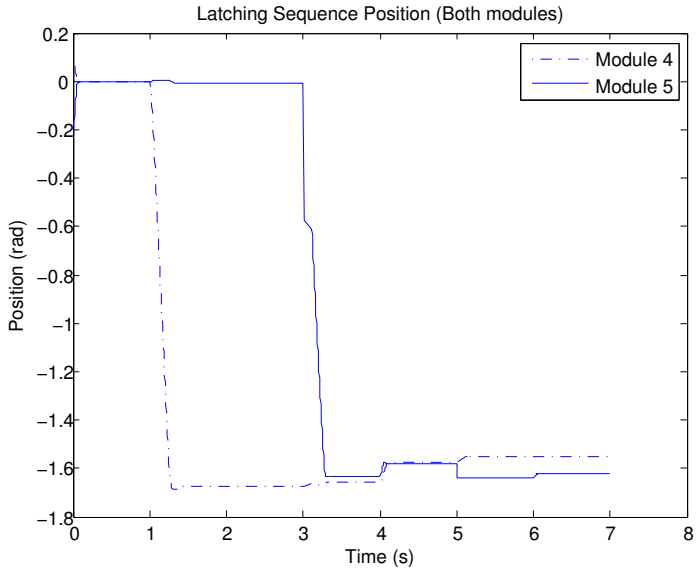


Figure 6.6: Latching sequence positions of modules 4 and 5.

connector are presented and explained, along with the corresponding quantitative measurements. Force testing to determine the impact of design parameters was presented, along with analysis of the corresponding behaviors. Bonding ratios were obtained for connector prototypes in the range from 20.6 to 79.1. While ABS plastic was the only material tested, different materials such as metals with sufficient elastic regions may yield much higher bonding ratio.

Reconfiguration on a 2D CKBot chain was performed and demonstrated to show simple attachment as well as progressive attachment of multiple connectors. Reconfiguration also demonstrated capabilities for lattice-style as well as chain-style

attachment. Finally, the latching control sequence taking advantage of the natural forces present in the connector was presented.

Chapter 7

Conclusion and Future Work

Connector performance is essential to docking, a core function of modular robotic systems. This dissertation presents multiple evaluation methods for self-aligning geometry to aid in finding high-performance designs. With the aid of the evaluation systems in this dissertation, we are able to design passive mechanical alignment connectors to reduce or remove elements such as actuation, sensing, etc., that would otherwise burden the design constraints of the system. These evaluation methods benefit from being neutral to system configuration and physical parameters.

7.1 Contributions

The new evaluation methods presented in this dissertation are defined over spaces which we call **areas of acceptance** for these connectors - a new concept which enumerates the limits of their passive error correction capabilities. These areas of acceptance come in many forms depending on the space of errors under examination and can be measured by either a Full or Oriented N-Cube metric. The methods we have detailed in this dissertation allow for enumeration of the acceptance for the first time in multiple dimensions simultaneously. These methods have been applied to connectors in this work up to the set of SE(3).

We have detailed a direct kinematic analysis method for incrementally and pseudo-statically moving a set of 2D connectors from the unaligned position to a final position. This method worked suitably well, but was ultimately very slow and has difficulty with more than one point of contact. Next we applied direct physical simulation with engines like ODE and Bullet. This works reasonably well at higher dimensions, but is even slower than the previous method, generally on the order of days.

By using the concept of configuration space and constructing 'obstacle' shapes based on connector geometry, we get a high-dimensional geometric model corresponding to the set of points where the two connectors are in contact. We presented a pre-partitioning method using polyhedral operations to analyse the surface of this shape and find the watershed corresponding to the area of acceptance, reducing the time required from days to hours.

By taking this method one step further and discretizing the obstacle shape, we were able to apply a 'flooding' algorithm from computer vision to find the watershed corresponding to the area of acceptance even faster. The discrete method reduces the amount of computational time required to get an answer by two orders of magnitude over the pre-partitioning method, allowing full-dimensional answers on the order of minutes. This computational savings also makes it practical to perform the analysis in more than 3 dimensions, a first in this kind of analysis. This is a leap forward in terms of methods to evaluate self-aligning connector pairs and determine their acceptance/error correction capabilities numerically. The results of these methods were verified on multiple robot platforms, in order to show that they have real-world applicability.

In order to consider also the method of physical connection after alignment, we also present a 2D instance of a high acceptance self-aligning connector with compliant latching for reconfiguration of a 2D modular chain robot in the plane. This connector was designed, constructed, implemented, and tested with an existing modular robot

system. along with force testing results and a latching command sequence.

It is our hope that these methods will be useful for the future design of robotic connectors and enhance the capabilities of modular robotic systems.

7.2 Future Work

In future, we would like to see more complex analysis as it pertains to real connectors, meaning the addition of friction, dynamic effects, etc. The addition of friction would be as simple as not allowing a point on the watershed to 'flow' downhill unless it meets a certain steepness of slope requirement - a gentle slope represents less contact forces in the directions of offset, meaning friction could cause a complete lack of motion in these cases. Dynamic effects would be trickier to manage, as the way in which one contact proceeds to another becomes less predictable, making the watershed analyses less helpful. Nonetheless, if the effects can be characterized as a probability, it may be possible to say the likelihood the pieces will land within the capture region, thus giving a successful dock.

Additionally, future work would include a more comprehensive look at connector geometries, testing multiple parametrically defined geometries in an attempt to find a definitive maximum for a specific use case. If a 3d connector is characterized as a height function of 2 variables, that function could be altered repeatedly and tested to find a 'best case' (possibly even using learning techniques), although I suspect the geometries presented within this thesis are very close to the maximum.

It is also possible that we could maximize the area of acceptance by changing the configuration space geometry and then reversing the Minkowski operation to find the resultant connectors. This would require one connector geometry assumed and perhaps a loosening of the requirement that they mate cleanly along a shared face, but nonetheless holds promise for potential future configurations of connectors that would improve alignment performance.

The docking process is not complete without physical connection. While the latching connector presented herein serves as a proof-of-concept for 2D cases, real systems usually require some method of undocking, perhaps by shape memory alloy or other low-profile actuation. A method of creating a passive physical connection (latching or otherwise) for 3D connectors would also be highly desirable for many robotic applications, though this would be potentially be more difficult to fabricate than the 2D latching connector.

A common issue when fabricating components is design for tolerancing; it can be assumed that any given dimension will very likely be off by some fraction, with a random distribution. Selecting the appropriate tolerances is a common design challenge [11]. This fabrication error can contribute to positioning errors within the system, though it can generally be reduced at increased cost. Since we are seeking to correct for positioning errors during docking, there could be a cost savings in terms of mechanical tolerances given a large enough area of acceptance. Future work might include an analysis of how relaxed tolerancing could ultimately result in cost savings for robotic systems with high self-alignment.

The discretized watershed method holds promise for other applications beyond docking - for example this serves as an easy way to determine the stability of a grasp as proposed by Rimon et al. [55]. It could also be used to analyse nearly any kind of contact mechanics between two objects such as for caging, pushing an object, using the environment to adjust object position, etc.

Finding the depth threshold is tricky and often has to be done manually by inspecting the values given. I believe it would be worth investigating hierarchical segmentation [6] to see if this step can be automated in that way.

Bibliography

[1] Rethink robotics website. `https://www.rethinkrobotics.com/sawyer/`. Accessed: 2019-04-29.

[2] T. F. Allen, E. Rimon, and J. W. Burdick. Two-finger caging of 3d polyhedra using contact space search. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2005–2012, May 2014.

[3] Mircea Badescu and Constantinos Mavroidis. Novel active connector for modular robotic systems. *Mechatronics, IEEE/ASME Transactions on*, 8(3):342–351, 2003.

[4] Mato Baotić. Polytopic computations in constrained optimal control. *AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 50(3-4):119–134, 2009.

[5] Richard Barnes, Clarence Lehman, and David Mulla. Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers & Geosciences*, 62:117–127, 2014.

[6] Serge Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical morphology and its applications to image processing*, pages 69–76. Springer, 1994.

[7] Serge Beucher and Fernand Meyer. The morphological approach to segmentation: the watershed transformation. *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 34:433–433, 1992.

[8] André Bleau and L Joshua Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77(3):317–370, 2000.

[9] Zack Butler and Daniela Rus. Distributed planning and control for modular robots with unit-compressible modules. *The International Journal of Robotics Research*, 22(9):699–715, 2003.

[10] Andres Castano, Alberto Behar, and Peter M Will. The conro modules for reconfigurable robots. *Mechatronics, IEEE/ASME Transactions on*, 7(4):403–409, 2002.

[11] Kenneth W Chase and William H Greenwood. Design issues in mechanical tolerance analysis. *Manufacturing Review*, 1(1):50–59, 1988.

[12] Heping Chen, Jianjun Wang, George Zhang, Thomas Fuhlbrigge, and Soenke Kock. High-precision assembly automation based on robot compliance. *The International Journal of Advanced Manufacturing Technology*, 45(9-10):999, 2009.

[13] Namik Ciblak and Harvey Lipkin. Remote center of compliance reconsidered. In *Proceedings of the ASME International Design Engineering Technical Conferences*, 1996.

[14] Namik Ciblak and Harvey Lipkin. Design and analysis of remote center of compliance structures. *Journal of Field Robotics*, 20(8):415–427, 2003.

[15] Namik Ciblak and Harvey Lipkin. Design and analysis of remote center of compliance structures. *Journal of robotic systems*, 20(8):415–427, 2003.

[16] J. Davey, J. Sastra, M. Piccoli, and M. Yim. Modlock: A manual connector for reconfigurable modular robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3217–3222, 2012.

[17] Jay Davey, Ngai Kwok, and Mark Yim. Emulating self-reconfigurable robots-design of the smores system. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4464–4469. IEEE, 2012.

[18] Jay Davey, Jimmy Sastra, Matt Piccoli, and Mark Yim. Modlock: A manual connector for reconfigurable modular robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3217–3222. IEEE, 2012.

[19] Nick Eckenstein and Mark Yim. Design, principles, and testing of a latching modular robot connector. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2012.

[20] Nick Eckenstein and Mark Yim. The x-face: An improved planar passive mechanical connector for modular self-reconfigurable robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3073–3078. IEEE, 2012.

[21] Nick Eckenstein and Mark Yim. Area of acceptance for 3d self-aligning robotic connectors: Concepts, metrics, and designs. In *Robotics and Automation, 2014. ICRA 2014. Proceedings of the 2014 IEEE International Conference on*. IEEE, 2014 (in submission).

[22] Nick Eckenstein and Mark Yim. Modular reconfigurable robotic systems: Lattice automata. In *Robots and Lattice Automata*, pages 47–75. Springer, 2015.

[23] Nick Eckenstein and Mark Yim. Modular robot connector area of acceptance from configuration space obstacles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3550–3555. IEEE, 2017.

[24] Nick Eckenstein and Mark Yim. Discrete configuration space methods for determining modular connector area of acceptance in higher dimensions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 430–435. IEEE, 2018.

[25] Guoqiang Fu, Arianna Menciassi, and Paolo Dario. Development of a genderless and fail-safe connection system for autonomous modular robots. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 877–882. IEEE, 2011.

[26] Toshio Fukuda, Seiya Nakagawa, Yoshio Kawauchi, and Martin Buss. Structure decision method for self organising robots based on cell structures-cebot. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 695–700. IEEE, 1989.

[27] Ricardo Franco Mendoza Garcia, Jonathan D Hiller, Kasper Stoy, and Hod Lipson. A vacuum-based bonding mechanism for modular robotics. *Robotics, IEEE Transactions on*, 27(5):876–890, 2011.

[28] Kyle Gilpin, Ara Knaian, and Daniela Rus. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2485–2492. IEEE, 2010.

[29] Kyle Gilpin, Kent Koyanagi, and Daniela Rus. Making self-disassembling objects with multiple components in the robot pebbles system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3614–3621. IEEE, 2011.

[30] Larry L Howell and A Midha. A method for the design of compliant mechanisms with small-length flexural pivots. *Journal of Mechanical Design*, 116(1):280–290, 1994.

[31] Larry L Howell, Ashok Midha, and TW Norton. Evaluation of equivalent spring stiffness for use in a pseudo-rigid-body model of large-deflection compliant mechanisms. *Journal of Mechanical Design*, 118(1):126–131, 1996.

[32] Mustafa E Karagozler, Brian Kirby, Wei Jie Lee, Eugene Marinelli, Tze Chang Ng, Michael P Weller, and Seth C Goldstein. Ultralight modular robotic building blocks for the rapid deployment of planetary outposts. 2006.

[33] Keith Kotay, Daniela Rus, Marsette Vona, and Craig McGray. The self-reconfiguring robotic molecule. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 424–431. IEEE, 1998.

[34] Haruhisa Kurokawa, Kohji Tomita, Akiya Kamimura, Shigeru Kokaji, Takashi Hasuo, and Satoshi Murata. Distributed self-reconfiguration of m-tran iii modular robotic system. *The International Journal of Robotics Research*, 27(3-4):373–386, 2008.

[35] Michael Dennis Mays Kutzer, Matthew S Moses, Christopher Y Brown, David H Scheidt, Gregory S Chirikjian, and Mehran Armand. Design of a new independently-mobile reconfigurable modular robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2758–2764. IEEE, 2010.

[36] Michal Kvasnica, Pascal Grieder, Mato Baotić, and Manfred Morari. Multi-parametric toolbox (mpt). In *Hybrid systems: computation and control*, pages 448–462. Springer, 2004.

[37] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[38] Sangcheol Lee. Development of a new variable remote center compliance (vrcc) with modified elastomer shear pad (esp) for robot assembly. *IEEE Transactions on Automation Science and Engineering*, 2(2):193–197, 2005.

[39] Qiao Lin, Joel Burdick, and Elon Rimon. A quality measure for compliant grasps. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 86–92. IEEE, 1997.

[40] Tomas Lozano-Perez. Spatial planning: A configuration space approach. In *Autonomous robot vehicles*, pages 259–271. Springer, 1990.

[41] Tomas Lozano-Perez, Matthew T Mason, and Russell H Taylor. Automatic synthesis of fine-motion strategies for robots. *The Int. Journal of Robotics Research*, 3(1):3–24, 1984.

[42] Alan P Mangan and Ross T Whitaker. Partitioning 3d surface meshes using watershed segmentation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):308–321, 1999.

[43] Richard Mason, Elon Rimon, and Joel Burdick. The stability of heavy objects with multiple contacts. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 439–445. IEEE, 1995.

[44] Luis A Mateos and Markus Vincze. Lammos-latching mechanism based on motorized-screw for reconfigurable robots.

[45] Fernand Meyer. Un algorithme optimal de ligne de partage des eaux. In *Dans 8me congres de reconnaissance des formes et intelligence artificielle*, volume 2, pages 847–859, 1991.

[46] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji. A 3-d self-reconfigurable structure. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 432–439 vol.1, 1998.

[47] Satoshi Murata, Haruhisa Kurokawa, and Shigeru Kokaji. Self-assembling machine. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 441–448. IEEE, 1994.

[48] Martin Nilsson. Symmetric docking in 2d: A bound on self-alignable offsets. In *IASTED '99: Robotics and Automation*, Oct. 1999.

[49] Martin Nilsson. Heavy-duty connectors for self-reconfiguring robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4071–4076. IEEE, 2002.

[50] Ian O'Hara, James Paulos, Jay Davey, Nick Eckenstein, Neel Doshi, Tarik Tosun, Jon Greco, Jungwon Seo, Matt Turpin, Vijay Kumar, and Mark Yim. Self-assembly of a swarm of autonomous boats into floating structures. In *Robotics and Automation, 2014. ICRA 2014. Proceedings of the 2014 IEEE International Conference on*. IEEE, 2014 (in submission).

[51] Esben Hallundbæk Østergaard, Kristian Kassow, Richard Beck, and Henrik Hautop Lund. Design of the atron lattice-based self-reconfigurable robot. *Autonomous Robots*, 21(2):165–183, 2006.

[52] Raymond Oung and Raffaello D'Andrea. The distributed flight array. *Mechatronics*, 21(6):908–917, 2011.

[53] Amit Pamecha, Imme Ebert-Uphoff, and Gregory S Chirikjian. Useful metrics for modular robot motion planning. *Robotics and Automation, IEEE Transactions on*, 13(4):531–545, 1997.

[54] Elon Rimon and Joel W Burdick. A configuration space analysis of bodies in contact—i. 1st order mobility. *Mechanism and Machine Theory*, 30(6):897–912, 1995.

[55] Elon Rimon and Joel W Burdick. A configuration space analysis of bodies in contact—ii. 2nd order mobility. *Mechanism and machine theory*, 30(6):913–928, 1995.

[56] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1, 2):187–228, 2000.

[57] John W. Romanishin, Kyle Gilpin, and Daniela Rus. M-blocks: Momentum-driven, magnetic modular robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3073–3078. IEEE, 2013.

[58] Ariel Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.

[59] Wei-Min Shen, Robert Kovac, and Michael Rubenstein. Singo: a single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4253–4258. IEEE, 2009.

[60] Masahiro Shimizu, Akio Ishiguro, and Toshihiro Kawakatsu. Slimebot: A modular robot that exploits emergent phenomena. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2982–2987. IEEE, 2005.

[61] Alexander Sprowitz, Soha Pouya, Stéphane Bonardi, Jesse Van den Kieboom, Rico Mockel, Aude Billard, Pierre Dillenbourg, and Auke Jan Ijspeert. Roombots: reconfigurable robots for adaptive furniture. *Computational Intelligence Magazine, IEEE*, 5(3):20–32, 2010.

[62] Stanislav L Stoev and Wolfgang Straβer. Extracting regions of interest applying a local watershed transformation. In *Proceedings of the conference on Visualization'00*, pages 21–28. IEEE Computer Society Press, 2000.

[63] Kasper Støy. *An Introduction to Self-Reconfigurable Robots*. MIT Press, Boston, MA, 2009.

[64] Kasper Støy. *An Introduction to Self-Reconfigurable Robots*. MIT Press, Boston, MA, 2009.

[65] J.W. Suh, S.B. Homans, and M. Yim. Telecubes: mechanical design of a module for self-reconfigurable robotics. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 4, pages 4095–4101 vol.4, 2002.

[66] Yiyong Sun, David Lon Page, Joon Ki Paik, Andreas Koschan, and Mongi A Abidi. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages 825–828. IEEE, 2002.

[67] Cem Unsal, Han Kiliccote, and Pradeep K Khosla. I (ces)-cubes: a modular self-reconfigurable bipartite robotic system. In *Photonics East'99*, pages 258–269. International Society for Optics and Photonics, 1999.

[68] Iuliu Vasilescu, Paulina Varshavskaya, Keith Kotay, and Daniela Rus. Autonomous modular optical underwater robot (amour) design, prototype and feasibility study. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1603–1609. IEEE, 2005.

[69] Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):583–598, 1991.

[70] Paul J White, Chris E Thorne, and Mark Yim. Right angle tetrahedron chain externally-actuated testbed (ratchet): A shape-changing system. In *Proc. Int. Design Eng. Tech. Conf. Comput. Inform. in Eng. Conf. IDETC/CIE*, volume 7, pages 807–817, 2009.

[71] Walton R Williamson, Eric Reed, Gregory J Glenn, Stephen M Stecko, Jeffrey Musgrave, and John M Takacs. Controllable drogue for automated aerial refueling. *Journal of Aircraft*, 47(2):515–527, 2010.

[72] Kevin D Wise and Adrian Bowyer. A survey of global configuration-space mapping techniques for a single robot in a static environment. *The Int. Journal of Robotics Research*, 19(8):762–779, 2000.

[73] M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan, and C.J. Taylor. Towards robotic self-reassembly after explosion. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2767–2772, 2007.

[74] Mark Yim, Kimon Roufas, David Duff, Ying Zhang, Craig Eldershaw, and Sam Homans. Modular reconfigurable robots in space applications. *Autonomous Robots*, 14(2-3):225–237, 2003.

[75] Mark Yim, Ying Zhang, Kimon Roufas, David Duff, and Craig Eldershaw. Connecting and disconnecting for chain self-reconfiguration with polybot. *Mechatronics, IEEE/ASME Transactions on*, 7(4):442–451, 2002.

[76] Victor Zykov, Andrew Chan, and Hod Lipson. Molecubes: An open-source modular robotics kit. In *Proc. IROS*, volume 7, 2007.